University of Modena and Reggio Emilia
XXXV cycle of the International Doctorate School in
Information and Communication Technologies

Rehearsal-Based Methods for Continual Learning

Matteo Boschini

Supervisor: Prof. Simone Calderara
Ph.D. Course Coordinator: Prof. Sonia Bergamaschi

Modena, 2023

Review committee:

Prof. Concetto Spampinato, University of Catania
Rahaf Aljundi, Ph.D., Toyota Motor Europe

To Francesca, Michele.

Abstract

Artificial Neural Networks (ANNs) have been established as the centrepiece of contemporary Artificial Intelligence, steadily raising the bar for what can be accomplished by computer programs thanks to their effectiveness and versatility. While they shine especially for their capability for generalisation, these systems impose the strict requirement that their training procedure should insist on independent and identically distributed data. In contrast with human intelligence – which seamlessly allows us to acquire knowledge continuously – ANNs forget previously acquired knowledge catastrophically whenever their training data distribution changes over time. Such a fundamental limitation prevents the development of intelligent systems capable of quick adaptation, crucially tying model updates to a cumbersome offline retraining procedure.

Continual Learning (CL) is a rapidly growing area of machine learning whose aim is counteracting the catastrophic forgetting phenomenon in ANNs through purposefully designed approaches. Among these, a prominent role is played by Rehearsal-Based Methods (RBM), which operate by storing few pieces of previously encountered data for later re-use, thus striking a favourable balance between efficacy and efficiency.

This thesis encompasses the contributions to CL made by the candidate during his doctoral studies. Starting from a review of recent literature, it highlights the relevance of RBMs and shows that the decades-old Experience Replay baseline is competitive with current state-of-the-art approaches when carefully trained. Subsequently, this manuscript focuses on the proposal of novel RBMs, which expand on the basic replay formula by leveraging knowledge distillation ([X-]DER), implicit dynamic adaptation of network capacity (LiDER) and geometric regularisation of the model's latent space (CaSpeR). Extensive experimental analyses highlight the merits of the proposed approaches, shedding light on the specific properties they confer on the in-training model.

Finally, this thesis investigates the applicability of RBMs beyond the typical incremental classification setting. Namely, a novel CL experimental scenario is introduced to provide more realistic evaluations w.r.t. common benchmarks in literature, an investigation is presented concern-

ing the viability of CL when limited supervision is available, a thorough study is conducted on the interplay between pre-training and CL. As a result, architectures and best practices are introduced that bridge the gap between standard CL evaluations and real-world applications.

Sommario

Le reti neurali artificiali (Artificial Neural Networks – ANN) hanno acquisito un ruolo di massimo rilievo nel contesto delle applicazioni contemporanee di Intelligenza Artificiale, portando ad un incremento costante delle potenzialità dei programmi informatici grazie alla loro efficacia e versatilità. Benché eccellenti nella loro capacità di generalizzazione, queste richiedono strettamente che il loro addestramento sfrutti dati indipendenti e identicamente distribuiti. Mentre l'intelligenza umana permette naturalmente di acquisire nuovi concetti in maniera incrementale, le ANN dimenticano la conoscenza pregressa in modo catastrofico ogniqualvolta intervenga una variazione nella distribuzione dei dati di addestramento. Questa limitazione fondamentale impedisce lo sviluppo di sistemi intelligenti capaci di adattarsi rapidamente al contesto in cui operano e vincola l'aggiornamento dei modelli a onerose procedure di riaddestramento.

L'apprendimento continuo (Continual Learning – CL) è una branca in rapido sviluppo del machine learning che si prefigge come obiettivo lo sviluppo di architetture volte a compensare la dimenticanza catastrofica nelle ANN. Tra le soluzioni proposte, un ruolo di primaria importanza è rivestito dai metodi rehearsal (Rehearsal-Based Methods - RBM), che evitano la necessità di riaddestramento mediante l'immagazzinamento e il riutilizzo una modica quantità di dati pregressi, individuando così un compromesso ottimale tra efficacia e efficienza.

Questa tesi raccoglie i contributi scientifici nell'ambito del CL prodotti dal candidato nel corso delle sue attività di dottorato. Inizialmente, si presenta un esame della letteratura recente, evidenziando la rilevanza degli RBM e mostrando che il noto approccio Experience Replay – proposto per la prima volta negli anni '90 – resta competitivo rispetto allo stato dell'arte quando si assumono opportuni accorgimenti operativi. Successivamente, il lavoro si focalizza sulla proposta di nuovi RBM che sfruttano i principi di distillazione di conoscenza ([X-]DER), adattamento dinamico implicito della capacità del modello (LiDER) e regolarizzazione geometrica dello spazio latente del modello (CaSpeR). Gli approcci proposti sono convalidati mediante estese analisi sperimentali, volte anche a mettere in risalto le specifiche proprietà da essi conferite al modello.

La parte finale di questa tesi presenta analisi dell'applicabilità di RBM a scenari che superano il tipico assetto sperimentale di classificazione incrementale: un nuovo esperimento volto a perseguire una modellazione più realistica dei cambi di distribuzione nei dati di ingresso, uno studio sulla applicabilità di CL in regime di supervisione limitata e una analisi sull'interazione tra CL e il pre-addestramento. Questi studi portano allo sviluppo di architetture e prassi operative volte a colmare il divario tra la letteratura e la applicazione di sistemi CL ad applicazioni realistiche.

Table of Contents

List of Abbreviations

A-GEM Averaged Gradient Episodic Memory

AI Artificial Intelligence

ANN Artificial Neural Network

BiC Bias Control

CaSpeR Continual Spectral Regulariser

CCIC Contrastive Continual Interpolation Consistency

CL Continual Learning

Class-IL Class-Incremental Learning

CO^{2}L Contrastive Continual Learning

CSSL Continual Semi-Supervised Learning

DER Dark Experience Replay

DER++ Dark Experience Replay++

DL Deep Learning

DNN Deep Neural Network

Domain-IL Domain-Incremental Learning

ECE Expected Calibration Error

EMA Exponential Moving Average

ER Experience Replay

ER-ACE Experience Replay with Asymmetric Cross-Entropy

ER-RPC Experience Replay with Regular Polytope Classifier

EWC Elastic Weight Consolidation

FA Final Accuracy

FAA Final Average Accuracy

FAAF Final Average Adjusted Forgetting

FAF Final Average Forgetting

FAIA Final Average Incremental Accuracy

FBWD Final Backward Transfer

FDR Function Distance Regularisation

FFWD Final Forward Transfer

FT Finetuning

GAP Global Average Pooling

GCL General Continual Learning

GDumb Greedy Sampler and Dumb Learner

GEM Gradient Episodic Memory

GSS Gradient-based Sample Selection

HAL Hindsight Anchor Learning

HAT Hard Attention to the Task

iCaRL Incremental Classifier and Representation Learning

JT Joint Training

LFR Local Flatness Regulariser

LGG Latent Geometry Graph

LiDER Lipschitz-Driven Experience Replay

LLR Local Linearity Regulariser

LUCIR Learning a Unified Classifier Incrmentally via Rebalancing

LwF Learning without Forgetting

LwF.MC MultiClass LwF

MER Meta-Experience Replay

ML Machine Learning

oEWC online Elastic Weight Consolidation

OLAP Online Structured Laplace Approximation

P-MNIST Permuted MNIST

P&C Progress & Compress

PackNet *Packing* Multiple Tasks into a Single Network

PNN Progressive Neural Networks

PODNet Pooled Outputs Distillation Network

QP Quadratic Programming

R-MNIST Rotated MNIST

RBM Rehearsal-Based CL Method

S-*mini*Img Sequential *mini*ImageNet

S-CIF10 Sequential CIFAR-10

S-CIF100 Sequential CIFAR-100

S-CORe50 Sequential CORe50

S-CUB200 Sequential CUB-200

S-FMNIST Sequential Fashion-MNIST

S-MNIST Sequential MNIST

S-NTU60 Sequential NTU-RGB+D-60

S-SVHN Sequential SVHN

S-TinyImg Sequential Tiny ImageNet

SGD Stochastic Gradient Descent

SI Synaptic Intelligence

SOTA state-of-the-art

SS-ERR Secondary-Superclass Error

SS-NLL Secondary-Superclass NLL

sSGD Stable SGD

Task-IL Task-Incremental Learning

TwF Transfer without Forgetting

X-DER eXtended Dark Experience Replay

X-DER $^{\text{CE}}_{\text{future}}$ X-DER with CE on future heads

X-DER $^{\text{no mem}}_{\text{update}}$ X-DER without memory update

X-DER $^{\text{RPC}}_{\text{future}}$ X-DER with RPC on future heads

X-DER $^{\text{w/out}}_{\text{future}}$ X-DER without future heads

Part I

Introduction

Chapter 1

Overview

1.1 Catastrophic Forgetting

Among the defining traits of human intelligence is the capacity to seamlessly and continually acquire new knowledge about the surrounding world; the human brain allows us to not only master novel and difficult tasks (*e.g.*, driving cars), but also to do so while remembering what was previously learnt (*e.g.*, riding a bicycle) without experiencing significant inference or forgetting. Artificial Neural Networks (ANNs) have recently come to represent an invaluable tool for allowing computer systems to perform similarly complex tasks on a variety of domains [56, 158, 170, 70]. We witness an unprecedented proliferation of Artificial Intelligence (AI) applications, bringing the general public in closer contact than ever with the quirks and peculiarities of these systems.

One fundamental limitation which clearly breaks the long-standing parallelism between the human brain and ANNs is the latter's dramatic failure on past data when trained on a stream of data whose distribution changes over time. The seminal studies by *McCloskey and Cohen* [109] and *Ratcliff* [137] highlight that this **catastrophic forgetting** effect is much more severe than the gradual forgetting curve observed in human test subjects [13] and that its root cause lies in the very structure of ANNs, whose shared set of weights is greedily optimised on current data through backpropagation (as illustrated in Fig. 1.1.

While initially measured on small and shallow models, catastrophic forgetting remains fundamentally relevant in modern Deep Neural Networks (DNNs) [49]. As it hinders the adaptation to mutating data, this phenomenon has a major impact on the life cycle of any modern AI application required to stand the test of time; developers and engineers must therefore periodically assess the performance of their deployed models and possibly carry out very expensive re-training and update procedures.

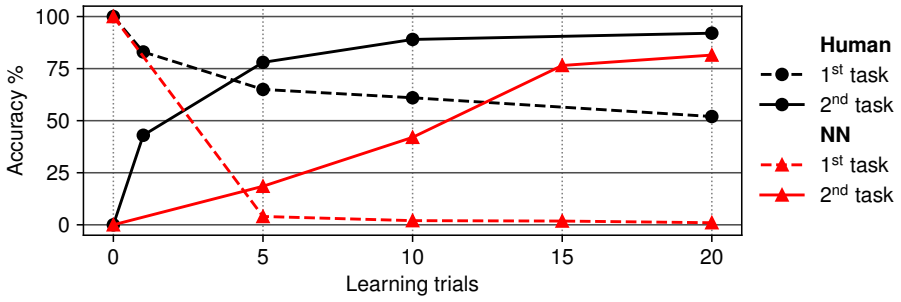


Fig. 1.1: Data from *Barnes and Underwood* [13] and *McCloskey and Cohen* [109]. A comparison between human and ANN performance on a two-task key-value learning experiment. While the 2$^{\text{nd}}$ task is being learnt, human subjects gradually forget the associations learnt in the 1$^{\text{st}}$ task; the same phenomenon is dramatically faster (*catastrophic*) in ANNs.

For this reason, the last few years have seen a renewed interest in the study of forgetting and development of solutions to reduce its incidence. The corresponding area in Machine Learning (ML) literature, typically labelled as **Continual Learning (CL)** [36, 127], has experienced swift growth in exposure at major AI conferences in recent years (see Fig. 1.2). Furthermore, industrial players also begin signalling strong interest on the subject, motivated by the possibility to achieve more robust and efficient operation of Deep Learning (DL) models [152, 74].

While several classes of CL approaches have been studied (a detailed breakdown will be presented in Sec. 2.5), one of the first proposed and most straightforward solutions to prevent catastrophic forgetting is given by **Rehearsal-Based CL Methods (RBMs)** [137, 143]. These operate by collecting some of the encountered data in a dedicated memory buffer and later reusing the gathered information to allow the model to keep training on past sample distributions. There are several factors making RBMs particularly appealing: *i)* Experience Replay (ER) [143, 28] – the simple baseline given by the repeated optimisation of in-memory items – is extremely easy to implement and already provides a very robust remedy to forgetting; *ii)* RBMs can easily be made to scale by adjusting the memory buffer size; *iii)* they are more forgiving to the learner, as the availability of past data can be used to revert forgetting after it has begun. Still, striving for effective exploitation of memorised data leads to challenging research questions concerning the characterisation, updating, management and usage of replay data.

[1]The reported statistics were obtained from a filtering of community-sourced CL paper list available at https://github.com/xialeiliu/Awesome-Incremental-Learning.

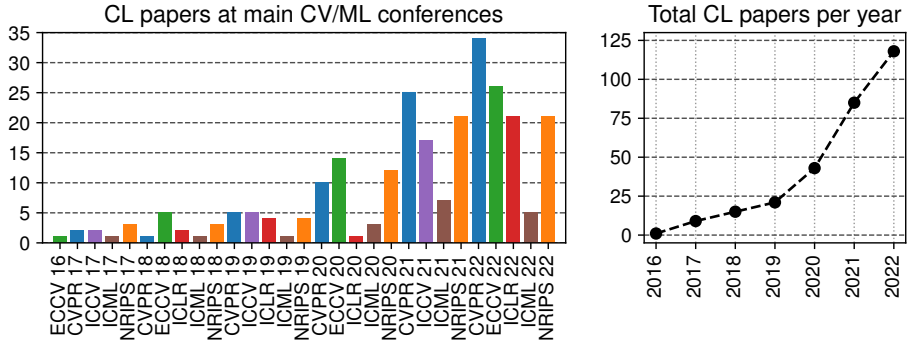


Fig. 1.2: The increase in popularity of CL: Count of CL-themed papers at major Machine Learning and Computer Vision conferences (left), overall amount of CL papers in major venues by year (right)[1].

1.2 Contributions and Organisation

This thesis encompasses the studies on CL conducted by the candidate throughout his doctoral studies. The contributions are organised as follows:

- **Part I** presents an **overview of the CL problem**.
  - this **Chap. 1** provides an overview of the problem of catastrophic forgetting and briefly introduces CL and RBMs.
  - **Chap. 2** formalises the CL classification problem, the experimental settings, benchmarks and evaluation metrics used in CL literature and, particularly, in this thesis. Finally, it presents a brief overview of the state of the art, with an emphasis on the approaches that will serve as competitors for the experiments presented in the following.
- **Part II** covers the **proposal of several novel RBMs**, each meant to address specific challenges in the incremental training of deep models.
  - **Chap. 3** initially frames the well-known Experience Replay (ER) RBM baseline in the context of modern literature. Subsequently, it highlights some possible pitfalls in the way ER is trained and proposes architectural modifications to improve its overall effectiveness.
  - **Chap. 4** introduces Dark Experience Replay (DER), a novel RBM that prevents forgetting by leveraging self-knowledge distillation, while also preserving the simplicity of ER. The chapter

includes a thorough evaluation of the proposed baseline against several competitors on multiple CL benchmarks and a detailed empirical analysis aiming at highlighting the core properties of the proposed method.

- **Chap. 5** re-evaluates DER and highlights some key limitations concerning its handling of secondary knowledge and its production of unified predictions. Consequently, an extended version of DER, called eXtended Dark Experience Replay (X-DER) is proposed; an extensive suite of analytical experiments is then presented, providing an in-depth analysis of the operation of X-DER.
- **Chap. 6** focuses on the uneven availability of input-stream and replay examples in RBMs and highlights that this determines overfitting of the latter to the detriment of the learner's decision boundary. To avoid this, it introduces Lipschitz-Driven Experience Replay (LiDER), a purposely designed regularisation term based on Lipschitz-constant regularisation. Through CL experiments and additional analysis, it is shown that LiDER can be beneficially combined with state-of-the-art (SOTA) RBMs.
- **Chap. 7** highlights that RBMs might fail at producing a disentangled latent space when trained incrementally. This effect can be averted by introducing a spectral-geometry motivated loss term called Continual Spectral Regulariser (CaSpeR). This leads to increased compactness on the latent space of SOTA RBMs, resulting in improved performance.

- **Part III** encompasses studies that go beyond typical CL experimental scenarios, meaning to bridge the gap between the theoretical study and the application of incremental learning systems.

  - **Chap. 8** proposes MNIST-360, a novel benchmark in line with the General Continual Learning (GCL) setting guidelines of [35] which features no clear task boundaries and requires the learner to deal with both sudden and gradual input distribution shifts. It is highlighted that only a small subset of methods in CL literature (all RBMs) are compatible with this setting.
  - **Chap. 9** introduces Continual Semi-Supervised Learning (CSSL), a scenario that relaxes the assumption – hard to meet in practical scenarios – that annotation all examples in the input stream can be timely annotated by a human supervisor. Doing so reveals that standard CL approaches are dramatically hindered and highlights the need for CSSL-specific approaches. In this regard, a first solution called Contrastive Continual

Interpolation Consistency (CCIC) is proposed along with an investigation into the applicability of the previously introduced LiDER in CSSL.

  - **Chap. 10** investigates the implication of the commonly adopted strategy of pre-training ML models when working in CL and highlights that the pre-training is itself subject to catastrophic forgetting and jeopardised in later tasks. A solution called Transfer without Forgetting (TwF) is then proposed to retain pre-training knowledge and facilitate the transfer of effective features while the model is trained continuously.

- Finally, **Part IV** wraps up the thesis by summarising the presented results and outlining potential future developments of CL and perspective research directions.

Chapter 2

Technical Background

2.1 Continual Learning Problem

While catastrophic forgetting can be studied in conjunction with any ML task (*e.g.*, segmentation [22, 189], detection [130, 69], generation [200], captioning [38], etc.), this thesis focuses on continual classification problems. Such a choice is in line with the majority of CL literature and allows us to highlight the key issues of incremental model operation with simple and easy-to-follow experiments. In the remainder of this manuscript, we will always use CL to indicate *continual learning classification* problems unless otherwise stated.

In CL, a model f with parameters θ is trained on a sequence of T tasks $\{\mathcal{T}_0, \dots, \mathcal{T}_{T-1}\}$. The i^{th} task consists of input-label pairs $\{x_i^{(n)}, y_i^{(n)}\}_{n=1}^{|\mathcal{T}_i|} \subset \mathcal{X}_i \times \mathcal{Y}_i$. While these data-points are *i.i.d.* within $\mathcal{T}_i$, the overall training procedure does not abide by the *i.i.d.* assumption, as the data distribution changes between tasks. The objective of CL is minimising the risk over all tasks:

$$\mathcal{L}_{\text{CL}} \triangleq \sum_{i=0}^{T-1} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{T}_i} \big[\mathcal{L}(f_\theta(x), y)\big], \tag{2.1}$$

where $\mathcal{L}$ indicates the loss associated with the classification task (*e.g.*, the categorical cross-entropy) given prediction $f_\theta(x)$ and ground-truth label y. While pursuing the optimal solution to Eq. 2.1 – $\theta^* = \text{argmin}_\theta \, \mathcal{L}_{\text{CL}}$ – the learner is not given free access to all data: only one task can be learnt at any given time and with a limited number of observations. To prevent the performance deterioration on past data associated with catastrophic forgetting, CL models combine the optimisation of the empirical risk on the current task $\mathcal{T}_c$ with a separate regularisation term $\mathcal{L}_R$:

$$\hat{\mathcal{L}}_{\text{CL}} \triangleq \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{T}_c} \big[\mathcal{L}(f_\theta(x), y)\big] + \mathcal{L}_{\text{R}}. \tag{2.2}$$

The additional term $\mathcal{L}_R$ can vary significantly for different models. This gives rise to distinct classes of CL approaches, which are presented in detail in Sec. 2.5.

2.2 Continual Learning Scenarios

The definition of the CL classification problem provided in the previous section is general enough to allow for the formulation of different experimental settings, with varying characteristics and degrees of complexity. As the field was initially characterised by subtle but critical differences in the way models were evaluated, early CL works did not allow for a straightforward comparison of results presented in different papers, even when referring to the same datasets. A rigorous taxonomy of possible experimental CL designs was then introduced by *Van de Ven et al.* [168], encompassing the three so-called *academic* scenarios. While this gave practitioners and researchers a common language for the description of CL experiments, these settings are often criticised for their failure to model key aspects of realistic incremental applications [5, 36]. As a result, additional *modern* scenarios have been proposed, which are meant to be more realistic and challenging by imposing further restrictions on what models are allowed to do while learning. Both kinds of settings will be covered in detail in this section.

For the sake of comparability with the majority of works in literature, the novel RBMs proposed in Part II of this thesis are evaluated on the former standardised settings. Conversely, Part III comprises studies of non-standard CL settings, possibly described as additional *modern* scenarios.

2.2.1 Academic Scenarios

All three *academic* scenarios in [168] present the model with a supervised classification problem split into tasks which are observed sequentially. During each task, the learner can only access a portion of the overall dataset; the change of task is notified to the model so that specific steps may be taken prior to moving on to the next task. Depending on the specific decision function that must be learnt, *Van de Ven et al.* distinguish the following:

- **Task-Incremental Learning (Task-IL)** presents the model with disjoint classification tasks (*i.e.*, $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$, with $\mathcal{Y}_i$ the set of classes shown in task $\mathcal{T}_i$) and – upon evaluation – always clarifies the task that must be recalled. Typically, this means that the model additionally receives a data-point's task identity $t \in \mathcal{J}$ during its forward step, *i.e.*, it must learn a task-conditioned classification function $f : \mathcal{X} \times \mathcal{J} \to \mathcal{Y}$.

Since this facilitates the disentanglement of knowledge belonging to distinct tasks and easily leads to saturated performance, Task-IL has been identified as the easiest of the classical scenarios [45, 6]. Still, this setting is particularly relevant for the quantification of forgetting occurring in the model, as its results are unaffected by the classifier bias linked to imbalanced data presentation [182];

- **Domain-Incremental Learning (Domain-IL)** presents all classes to the model during each task, making the input subject to a task-dependent transformation. Task identities need not be made available at test time: all classes are shown during each task and the model must provide a prediction irrespective of the task it is observing, *i.e.*, the model must learn a task-independent classification function $f : \mathcal{X} \rightarrow \mathcal{Y}$. Domain-IL experiments require the employment of dedicated datasets either obtained by applying classes of transformations on top the data-points of classification datasets (*e.g.*, for image classification, rotations, pixel permutations, etc. applied on top of MNIST [88], CIFAR [83], etc.) or already comprising of multiple domains (*e.g.*, the DomainNet dataset, originally employed for Domain Adaptation Problems [129]);

- **Class-Incremental Learning (Class-IL)** consists of a disjoint classification problem like Task-IL; however, the task identity is not provided at test time, meaning that the model must learn a classification function identifying both the source task and class within it $f : \mathcal{X} \rightarrow \mathcal{J} \times \mathcal{Y}$. The learner is thus presented with the additional challenge to incrementally learn a unified classifier [60], which may problematically lead to the accumulation of bias in favour of the currently seen classes [182, 3]. As this is regarded as the most difficult and meaningful of the three scenarios [45], it is followed by the majority of the evaluations in the following of this manuscript.

2.2.2 Novel Scenarios

Although the *academic* scenarios operate a clear distinction w.r.t. the problem that will be solved by the online learner, they leave some degree of freedom regarding crucial factors in the way the experimental setting is practically constructed. For this reason, novel scenarios have been proposed which are gaining significant traction in current CL literature. We list the most relevant proposals in the following:

- **Online Continual Learning** is generally based on either Task-IL or Class-IL and introduces the additional requirement that data-points can only be shown once to the model, on the basis that any real-world continual learner would never be subject to the same input twice [105, 141, 26]. For this reason, it forbids performing multiple training

epochs on any task. While this is not problematic for datasets including a large number of examples per class (*e.g.*, MNIST), it can lead to underfitting if the opposite is true (*e.g.*, CIFAR-100) [53]. Most recently, *Caccia et al.* further proposed an extension of this setting by requiring that the model should be possibly evaluated at any time during training (*anytime evaluation*) [19];

- **Task-Free Learning** (sometimes also referred to as *Data-Incremental Learning*) further expands the Class-IL setting by mandating that task identities should not be made available even at training time [5, 67, 177, 134, 36, 157]. Similarly to what happens if an incremental learner is deployed on an in-the-wild stream of data, this setting allows a task change to occur at arbitrarily at any point in time and the model is not given the chance to prepare for the task boundary by carrying out additional operations;

- **Data-Free Class-Incremental Learning** is also defined on top of Class-IL, but instead questions the opportunity of storing and replaying previous examples [193, 161, 99, 47]. This specification is usually vaguely motivated by alleged privacy concerns and effectively disqualifies all RBMs from evaluation. As the latter consistently achieve the highest accuracy values in all other settings, this scenario effectively gives practitioners the means to further the study of non-replay CL methods without focusing too much on performance.

The novel CL settings introduced in Part III of this thesis also go beyond the *academic* settings by specifying additional experimental requirements. Specifically, the General Continual Learning (GCL) setting proposed in Chap. 8 can be seen as a special case of Task-Free Learning, with the additional introduction of a concurrent soft domain shift in addition to the existing hard class shift; the Continual Semi-Supervised Learning (CSSL) setting proposed in Chap. 9 rejects the hypothesis of full supervision being available for all data-points on the input stream, limiting annotations to just a few per class. The experiments in Chap. 10 formally adhere to the Class-IL and Task-IL, but explicitly assume that the employed models are pre-trained at the beginning of CL and investigate the consequences of this hypothesis.

2.3 Continual Learning Benchmarks

This section outlines the main characteristics of the CL benchmarks used in the remainder of this thesis. We report a brief overview in Tab. 2.1; unless otherwise stated, all experiments presented in the following chapters will abide by these specifications, adopt Stochastic Gradient Descent (SGD) as an optimiser and employ a validation set consisting of

10% of the training data for the purpose of hyper-parameter tuning. All of the presented benchmarks refer to image classification tasks, with the sole exception of S-NTU60, which involves action classification.

- **Sequential MNIST (S-MNIST)** [168] is obtained by splitting the classical MNIST image classification dataset [88] into 5 tasks with 2 classes each. While S-MNIST is a simple and well-known benchmark, it has been criticised as not fully representative of modern CV tasks [184, 45]. Each task includes approximately 6000 training images and 1000 test images;
- **Sequential Fashion-MNIST (S-FMNIST)** [30] consists of 5 tasks featuring 2 classes each, with 6000 and 1000 28×28 images per class. It is based on the Fashion-MNIST dataset [184] which was designed as a drop-in replacement for MNIST, allowing for the seamless repurposing of MNIST-based models on a more challenging benchmark;
- **Sequential CIFAR-10 (S-CIF10)** [199] is organised in 5 tasks with 2 classes each and 5000 and 1000 32×32 RGB images per class for training and testing respectively, coming from the CIFAR-10 dataset [83];
- **Sequential CIFAR-100 (S-CIF100)** [199, 138, 28] is obtained by splitting the CIFAR-100 dataset [83] into 10 consecutive tasks, each comprising of 10 classes with 500 and 100 32×32 images each for training and testing respectively;
- **Sequential CORe50 (S-CORe50)** [108] is comprised of 50 classes of common household items pictured at 128×128 resolution on varying backgrounds, with around 2400 examples per class. We follow the SIT-NC protocol described in [108] and organise these classes in 9 tasks, the first of which includes 10 classes whereas the following ones 5 each;
- **Sequential Tiny ImageNet (S-TinyImg)** is obtained by splitting the Tiny ImageNet dataset [163] into 10 tasks with 20 classes each. Each image comes in RGB at a 64×64 resolution, each class includes 500 training images and 50 test images;
- **Sequential *mini*ImageNet (S-*mini*Img)** [28, 43, 40] is obtained from *mini*ImageNet [171], a 100-class subset of the popular ImageNet [39] dataset, split in 20 classification tasks. Each task presents 84×84 RGB images out of 5 disjoint classes, each class features 500 and 100 training and test images respectively;
- **Sequential CUB-200 (S-CUB200)** [44] derives from a sequential split of the Caltech-UCSD Birds-200 fine-grained visual classification dataset [174]. This benchmark includes a very limited amount of data,

with just 30 training and test images per class. For this reason, we always start from an ImageNet pre-trained model when working on it in our experiments;

- **Sequential SVHN (S-SVHN)** derives from a sequential split of the The Street View House Numbers (SVHN) Dataset [120]. Classes are not balanced and contain an average of approximately 7300 training images and 2600 test images;
- **Permuted MNIST (P-MNIST)** [81] is a Domain-IL classification task obtained by applying a random pixel permutation on MNIST [88] images. Each task therefore consists of approximately 60000 training images and 10000 test images;
- **Rotated MNIST (R-MNIST)** [105] is a Domain-IL classification task obtained by applying a random rotation on MNIST [88] images, with an angle in the $[0, \pi)$ interval. As in the case of P-MNIST, each task includes approximately 60000 training images and 10000 test images;
- **Sequential NTU-RGB+D-60 (S-NTU60)** is obtained by splitting into tasks the NTU-RGB+D action classification dataset [153]. Each input point consists of 3D-space coordinates referring to 25 body joints tracked for 300 frames on up to 2 subjects. We adopt the *cross-subject* data-split [153], reserving distinct subjects for train and test data, resulting in 40091 and 16487 training and validation samples respectively, *i.e.*, approx. 6600 and 2700 images for each of the 6 tasks (with 10 classes each).

In addition to the ones listed above, we introduce a novel benchmark called **MNIST-360** in Chap. 8. As detailed later, this is also obtained by transforming the basic MNIST dataset; however, it abides by the requirements of the newly proposed GCL setting and thus cannot be applied to any of the *academic* scenarios.

Dataset	Input Shape	Tasks	Classes per Task	Backbone	Epochs per Task	Batch Size	Decay Rate (epochs)	Data Aug.*
S-MNIST	28×28	5	2	2× Linear (hs=100)	1	10	no	no
S-FMNIST	28×28	5	2	2× Linear (hs=256)	1	10	no	H N
S-CIF10	$3 \times 32 \times 32$	5	2	ResNet-18	50	32	no	C H N
S-CIF100	$3 \times 32 \times 32$	10	10	ResNet-18	50	32	0.2 (35, 45)	C H N
S-CORe50	$3 \times 128 \times 128$	9	5†	ResNet-18	15	32	no	C H N
S-TinyImg	$3 \times 64 \times 64$	10	20	ResNet-18	100	32	no	C H N
S-*mini*Img	$3 \times 84 \times 84$	20	5	EfficientNet-B2	80	128	0.2 (35, 60, 75)	C H N
S-CUB200	$3 \times 224 \times 224$	10	20	ResNet-50	50	64	no	C H N
S-SVHN	$3 \times 32 \times 32$	5	2	3×Conv2D (chan=100)	10	32	no	C N
P-MNIST	28×28	20	10	2× Linear (hs=100)	1	128	no	no
R-MNIST	28×28	20	10	2× Linear (hs=100)	1	128	no	no
S-NTU60	$3 \times 300 \times 25 \times 2$	6	10	Efficient-GCN-B0	70	16	0.2 (10, 50)[1]	R G

Tab. 2.1: Reference table for the experimental benchmarks used in this thesis. † the first task of S-CORe50 includes 10 classes. * Augmentation types: **N** normalisation, **H** random horizontal flip, **C** random crop, **R** random rotation, **G** Gaussian noise.

2.4 Evaluation Metrics

As CL experiments span multiple tasks, several evaluation metrics have been proposed in literature to express distinctive aspects of the learning dynamics. Let a_i^t denote the model accuracy on the i$^{\text{th}}$ task after training on task $\mathcal{T}_t$, the following measures have been defined:

- **Final Average Accuracy (FAA)** assesses the final average performance on the overall joint classification problem after learning all tasks incrementally:

$$\text{FAA} \triangleq \sum_{j=0}^{T-1} a_j^{T-1}, \tag{2.3}$$

  where the $\sum$ symbol denotes the averaging operation. This measure provides a compact summary of the trade-off between learning a task in an incremental manner or doing so jointly by allowing a direct comparison with the *i.i.d.* baseline. For this reason, FAA is largely adopted in literature [105, 35, 6] and it is the main performance indicator used in this manuscript;

- **Final Average Incremental Accuracy (FAIA)** [138, 60] is an alternative formulation of FAA taking into account the accuracy at the end of each task, thus also providing a compact summary of the historic performance:

$$\text{FAIA} \triangleq \sum_{i=0}^{T-1} \sum_{j=0}^{i} a_j^i. \tag{2.4}$$

  This metric usually yields slightly higher values w.r.t. FAA and might lead to reader confusion if the difference is not clearly specified;

- **Final Average Forgetting (FAF)** [25, 27, 26] measures the average performance degradation occurring on past tasks between their peak and final accuracy:

$$\text{FAF} \triangleq \sum_{j=0}^{T-2} f_j, \quad \text{s.t.} \quad f_j = \max_{l \in \{0,\dots,T-2\}} a_j^l - a_j^{T-1}. \tag{2.5}$$

  This measure can also take on a negative value in the case of a model which improves its accuracy on past tasks over time (a phenomenon known as positive *backward transfer*);

- **Final Average Adjusted Forgetting (FAAF)** is an adjusted version of FAF that we first proposed in the original paper for the approach presented in Chap. 7. This variant aims at allowing an easier comparison of the forgetting rate between models with different

peak accuracy by focusing on performance degradation alone and excluding backward transfer:

$$\begin{aligned} \text{FAAF} &\triangleq \sum_{j=0}^{T-1} \left[\frac{a_j^* - a_j^{T-2}}{a_j^l} \right]^+, \\ \text{s.t. } a_j^l &= \max_{t \in \{j,\dots,T-1\}} a_j^t, \quad \forall j \in \{0,\dots,T-2\}, \end{aligned} \tag{2.6}$$

where $[\cdot]^+$ indicates lower-bound clipping to zero;

- **Final Backward Transfer (FBWD)** [105] is a measure of accuracy degradation that – opposite to FAF – accounts for the performance increase on previously-learnt classes:

$$\text{FBWD} \triangleq \sum_{j=0}^{T-2} a_j^{T-1} - a_j^j. \tag{2.7}$$

  Such a measure is typically relevant for Domain-IL settings, where knowledge about a class can be improved in hindsight. It is otherwise typically negative for Class-IL and Task-IL, provided that the classes presented in different tasks are sufficiently dissimilar;

- **Final Forward Transfer (FFWD)** [105] is somewhat complementary to FBWD in that it measures the model's performance improvement over yet-to-be-seen classes:

$$\text{FFWD} \triangleq \sum_{j=1}^{T-1} a_j^{j-1} - a_j^{\text{init}}, \tag{2.8}$$

  where a_j^{init} denotes the accuracy on task $\mathcal{T}_j$ of the randomly initialised model. A high FFWD indicates that the model is handling learning so as to maximise generalisation, by improving its accuracy on classes of the j^{th} task w.r.t. its initialisation.

2.5 State of the Art

In this section, we present a review of some of the most important CL approaches, which will serve as competitors in the experiments in the following of this work. CL methods are usually categorised in the three families [45, 35] which we present in the following subsections. Additionally, CL experiments usually express a lower and upper bound for the results by reporting two **baseline** approaches: **Finetuning (FT)** and **Joint Training (JT)**. The former consists of directly training the DNN on the incoming stream of data with no additional remedy to catastrophic forgetting; the latter instead trains the backbone model on all available data jointly and is therefore not subject to forgetting at all.

2.5.1 Architectural Methods

Architectural methods are typically very effective in counteracting forgetting, as they devote distinguished sets of parameters to distinct tasks. Most of them are characterised by a multi-headed architecture [147] or otherwise use the provided task information to map portions of the model to specific tasks [151]. In both cases, their use is limited to the Task-IL scenario, as their operation depends on the availability of task labels at prediction time. Among them, we mention the following:

- **Progressive Neural Networks (PNN)** [147] is the archetypal architectural method, in that it straightforwardly devotes a separate replica of the backbone network to each task. Additionally, it facilitates knowledge transfer from previously instantiated backbones to new ones by introducing dedicated adaptation layers, that serve as bridges between them. While this methodology avoids forgetting by design, it has a steep memory requirement which grows linearly with the number of tasks;

- ***Packing* Multiple Tasks into a Single Network (PackNet)** [107] manages the backbone model by identifying the important weights after each task and freezing them. The remaining parameters are pruned out and devoted to new tasks. This leads to a strict compartmentalisation of model weights, which define distinct sub-networks used in different tasks. At evaluation time, a prediction is made by only using those weights that are assigned to the target task;

- **Hard Attention to the Task (HAT)** [151] learns task-specific attention masks to sparsify the allocation of network parameters to any given task. By so doing, it achieves a similar behaviour to PackNet, but also allows a degree of weight sharing between tasks, as it employs soft masks.

2.5.2 Regularisation Methods

Regularisation-based methods do not alter the model's architecture, but instead condition its evolution by means of additional loss function terms to prevent forgetting previous tasks. Such terms typically entail constraints either on the model's response [96, 149] or on its parameters [81, 199, 142]. We list the following:

- **Learning without Forgetting (LwF)** [96] applies functional regularisation by applying knowledge distillation [59] between the in-training model and a previous snapshot taken at the last task boundary w.r.t. current training examples. A notable variant to this approach is

MultiClass LwF (LwF.MC), which adopts a slightly different loss function[1] and is designed to operate in the Class-IL setting [138];

- **Progress & Compress (P&C)** [149] is a second representative of the functional regularisation family which, unlike LwF, employs a separate backbone for the exclusive learning of the current task. At task boundaries, it distils the acquired knowledge into a unified *Knowledge Base* model, in charge of preserving information on all previous tasks;

- **Elastic Weight Consolidation (EWC)** [81] is a well-known approach based on the regularisation of those model parameters that are identified as particularly relevant for past tasks. This is accomplished by estimating the Fisher Information Matrix after each task, which is used as a per-weight importance measure in later learning. As this is a cumbersome and memory-expensive procedure, a more efficient approximation called **online Elastic Weight Consolidation (oEWC)** has also been proposed [149];

- **Synaptic Intelligence (SI)** [199] similarly prevents the change of parameters which are deemed important to previous tasks. However, instead of estimating such importance at the task boundary like EWC, it operates online by keeping track of the cumulative loss decrease which can be ascribed to each parameter;

- **Online Structured Laplace Approximation (OLAP)** [142] proposes a refinement on the parameter-importance approach by leveraging a sophisticated block-diagonal Kronecker factored approximation of the loss Hessian. This is in contrast with the simple diagonal approximation employed by EWC and thus allows keeping track of the interactions between different weights;

- **Stable SGD (sSGD)** [113] adopts a slightly different approach with respect to previous methods, as it introduces specific alterations to the model's training regime with the purpose of biasing the optimisation towards flat minima of the loss landscape.

2.5.3 Rehearsal-Based Methods

Rehearsal-Based Methods (RBMs) operate by maintaining a fixed-size working memory of previously encountered exemplars, which are then used to prevent forgetting by either replaying them directly and/or using them as an additional source of regularisation. As shown by the experiments in the following of this manuscript, RBMs are characterised by robust operation and usually outperform regularisation baselines on Domain-IL

[1] LwF.MC is effectively equivalent to iCaRL without any memory buffer.

and Class-IL experiments. Even though rehearsal could be suggestive of the biological operation of memory in the animal brain [21, 139], the development of CL models is mostly unrelated to any biological insights.

- **Experience Replay (ER)** [137, 143] is the most straightforward approach to rehearsal and among the first historically proposed approaches for countering catastrophic forgetting. In it, a small memory buffer is allocated containing an *i.i.d.* sample of all previously encountered training data. Usually, the buffer is populated through the well-known *reservoir sampling* [173], which operates online and assigns each item on the stream the same probability of being included in the memory. In later tasks, data from the stream is interleaved with data sampled from the buffer, allowing for a joint optimisation. ER remains a strong baseline and the basis for many SOTA approaches in current literature;

- **Gradient-based Sample Selection (GSS)** [6] introduces a more refined sampling algorithm w.r.t. reservoir by explicitly selecting those samples whose induced gradient on the model best approximates the overall gradient of the original task. By so doing, they provide a more robust anchor for keeping the backbone model within the feasible region of the learnt task;

- **Experience Replay with Regular Polytope Classifier (ER-RPC)** [132] complements ER with the **Regular Polytope Classifier** proposed in [131]. Such a classifier constrains the parameters of the final classification layer to have constant values, designed to keep them equally distributed in output space. This implicates that all seen and unseen classes in the classification problem are kept at equal distance;

- **Meta-Experience Replay (MER)** [141] complements ER with a meta-learning objective with the purpose of maximising transfer and minimising interference. In doing so, it employs the Reptile algorithm [122] to compute a candidate weight modification for each batch example and finally aggregates all proposed updates into a comprehensive across-batch update. As a consequence, MER effectively adopts a batch size of 1, which dramatically slows its operation compared to ER and other methods;

- **Incremental Classifier and Representation Learning (iCaRL)** [138] adopts a self-knowledge distillation loss term similar to LwF to prevent the learnt representations from deteriorating in later tasks. On top of this, its predictions leverage a prototype-based *nearest-mean-of-exemplars* classifier which compares the features of input examples with the mean feature produced by all buffer examples of each class, generally leading to robust predictions even with complex benchmarks and small memory buffers;

- **Experience Replay with Asymmetric Cross-Entropy (ER-ACE)** [19] identifies and addresses class imbalances in the predictions made by ER caused by the different availability of stream and buffer exemplars. By simply separating the cross-entropy contribution of the classes in the two data sources, the authors achieve increased accuracy and reduced interference;

- **Learning a Unified Classifier Incrmentally via Rebalancing (LUCIR)** [60] builds on top of iCaRL by proposing several modifications that promote separation in feature space and thus result in an incrementally learnt classifier that is less affected by the bias between current- and previous-task predictions;

- **Bias Control (BiC)** [182] similarly deals with the problem of imbalanced predictions in ER, but does so by means of an additional regularisation term resembling the objective of LwF and introducing a dedicated model layer that is trained after each task to equalise predictions so as to counteract bias;

- **Hindsight Anchor Learning (HAL)** [26] improves ER by learning a set of data-points which maximise forgetting and subsequently introducing a regularisation term which anchors network responses on them, so as to prevent interference on its weakest spots;

- **Function Distance Regularisation (FDR)** [15], similarly to LwF and iCaRL, employs a self-distillation loss term against responses at the task boundary to regularise the model in function space to prevent forgetting in later tasks;

- **Pooled Outputs Distillation Network (PODNet)** [42] expands on iCaRL by extending the self-distillation loss term to convolutional layers and allowing for multi-modal representations in the proxy-based classifier;

- **Contrastive Continual Learning (CO^2L)** [23] proposes to facilitate knowledge transfer from samples stored in the buffer by optimising a contrastive learning objective to avoid the potential bias introduced by a cross-entropy objective. However, a linear classifier needs to be first trained for the purpose of inference;

- **Gradient Episodic Memory (GEM)** [105], unlike previously listed approaches, does not use its memory buffer for rehearsal. Instead, it exploits its data by building one Quadratic Programming (QP) constraint per previous task which explicitly aims at minimising the interference between the gradient of previous-task data and current inputs;

- **Averaged Gradient Episodic Memory (A-GEM)** [27] proposes an efficient approximation of GEM, which replaces the multiple QP constraints with an easier-to-compute averaged objective;
- **Greedy Sampler and Dumb Learner (GDumb)** [135] is an experimental method that also deviates from the standard rehearsal formula. It avoids training on input data entirely, limiting itself to gathering data into the memory buffer for later use. When an evaluation is required, GDumb trains a new model on the memory buffer from scratch. As it manages to outperform several competitors, this approach questions the significance of recent advances in CL.

Part II

Novel Rehearsal Methods for Continual Learning

Chapter 3

Rethinking Experience Replay

3.1 Motivation

In this chapter, we assess the role of ER [137] as a viable Class-IL baseline method by comparing it with more recent SOTA RBMs. The focus on the latter class of methods derives from architectural methods being generally not applicable outside of the Task-IL setting – as discussed in Sec. 2.5 – and from the tendency of regularisation-based approaches to underperform when compared to RBMs on Class-IL [6, 45].

While recently proposed RBMs are often shown to outperform a simple replay baseline [6, 4, 141, 138, 26], they typically operate by introducing additional regularisation terms that imply an increase in computational requirements and memory complexity. On the contrary, ER presents a very straightforward formulation, produces a limited overhead w.r.t. FT, but is exposed to several issues when applied to the Class-IL setting:

(a) repeated rehearsal on a limited memory buffer produces **overfitting**[1];

(b) incrementally learning a classifier produces a **bias towards newer classes**, to the detriment of earlier tasks [182, 60];

(c) the typical **random sampling** procedures applied for buffer population [141, 28] can be prone to failure cases (*e.g.*, some classes may be left out when the buffer is small).

In the following, we illustrate how ER can be brought to a performance in line with contemporary RBMs by introducing a few modifications (a bag of *tricks*) that address the above-mentioned issues. As these can be easily applied to other models, this chapter generally aims at constituting a quick reference for improving the design practices of CL approaches.

[1] An early-sampled item in the S-CIF10 protocol (buffer size 500; replay batch size 32) is replayed approximately 5000 times.

3.2 Experience Replay

ER realises the generalised CL problem formulation of Eq. 2.2 by introducing a very simple additional regularisation term:

$$\mathcal{L}_{\mathrm{R}} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{M}} \Big[\mathrm{CE}\left(f_\theta(x), y\right) \Big], \tag{3.1}$$

where $\mathcal{M}$ denotes a memory buffer, *i.e.*, a small fixed-size storage that is used for saving encountered exemplars and labels from previous tasks. During each training step, ER merges some of these items with the current batch: consequently, the network rehearses past tasks as it learns current data and thus achieves an approximation of Eq. 2.1.

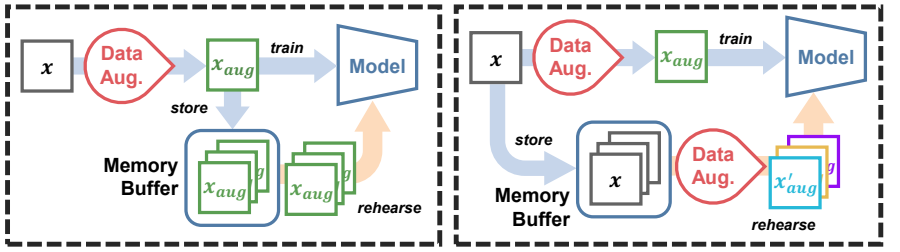
This practical solution only introduces two additional hyper-parameters to f_θ, namely the replay buffer size $|\mathcal{M}|$ and the number of elements that we draw from it at each step. As typically done in literature [141, 28], our baseline employs the *reservoir* sampling strategy [173] (see Alg. 3.1). For a balanced classification problem, this approach guarantees that each input exemplar has the same probability $|\mathcal{M}|/\sum_i |\mathcal{T}_i|$ of entering the replay buffer. We prefer this approach both to *herding* [138] and the *class-wise FIFO* [28] (a.k.a. *ring* buffer) strategies. Unlike *reservoir*, the former needs to retain the entire training set of each task; conversely, the latter fails to exploit the whole memory in earlier task and presents a higher risk of overfitting [28].

3.3 Training Tricks

In this section, we go over the details of some issues encountered by ER in the Class-IL setting. Consequently, we propose effective tricks to mitigate them by introducing some slight alterations in the base model. Specifically, Sec. 3.3.1, 3.3.4 and 3.3.5 propose improvements for the replay buffer (their extension to other RBMs is thus trivial). On the other hand, Sec. 3.3.2 and 3.3.3 present even more general tricks that can be applied to any Class-IL-capable CL method, as we show in Sec. 3.4.4.

3.3.1 Independent Buffer Augmentation (IBA)

Data augmentation is an obvious strategy for improving the generalisation capabilities of a DNN [180]. When dealing with CL scenarios, data augmentation is typically applied on the input stream of data from the current task. However, a RBM also learns from a second, more overfitting-prone source of data: its replay buffer. In addition to the regular augmentation performed on the input stream, we propose the adoption of Independent Buffer Augmentation (IBA). This requires storing



(a) (b)

Fig. 3.1: Graphical comparison between rehearsal on augmented examples (a) and Independent Buffer Augmentation (b).

non-augmented examples in the memory and subjecting them to independent data augmentation when drawn for later replay. This is done to introduce additional variety in the rehearsal examples and minimise overfitting on the memory.

While this may appear as a simple expedient, its application in literature should not be taken for granted. As an example, the CL methods implemented in the codebases of [105][2] and [6][3] store the augmented examples in the memory buffer and re-use them as-they-are, as illustrated in Fig. 3.1a. On the contrary, we remark that it is much more beneficial to show the model replay examples that undergo distinct transformations, as shown in Fig. 3.1b.

3.3.2 Bias Correction (BiC)

The sequential presentation of data within the Class-IL setting is known to give rise to a bias on model predictions, favouring classes from the current task [60, 182]. Such bias is not localised to its final classification layer, but rather linked to the whole model, meaning that the trivial solution of zeroing the classifier is not beneficial.

Hou et al. [60] address this issue structurally by devising a specific margin-ranking loss term aimed at keeping representations from different tasks separated. Instead, *Wu et al.* propose a much simpler and modular solution in [182], which we also apply here: the addition of a simple Bias Correction Layer to the model. This layer consists of two parameters α and β, used to compensate the k^{th} output logit $\ell_k \triangleq h_\theta(x)_k$, where we use h

[2] https://github.com/facebookresearch/GradientEpisodicMemory
[3] https://github.com/rahafaljundi/Gradient-based-Sample-Selection

to indicate pre-softmax model responses ($f_\theta(\cdot) \triangleq \text{softmax}\, h_\theta(\cdot)$), as follows:

$$q_k = \begin{cases} \alpha \cdot \ell_k + \beta & \text{if } k \text{ was trained in the last task} \\ o_k & \text{otherwise} \end{cases}. \tag{3.2}$$

This a layer is applied downstream of the classifier to yield the final output at test time. Thanks to its small size, it can be easily trained at the end of each task by leveraging a limited number of exemplars. While [182] employs a separate validation set for this purpose, we find it sufficient to exploit the same replay buffer we use for ER. Parameters α and β are optimised through the cross-entropy loss, as follows:

$$\mathcal{L}_{\text{BiC}} = -\sum_k \mathbb{1}_{y=k} \log[\text{softmax}(q_k)]. \tag{3.3}$$

3.3.3 Exponential LR Decay (ELRD)

It could be argued that not learning anything new is one of the best ways to preserve previous knowledge. To this aim, we propose to decrease the learning rate progressively at each iteration; we found exponential decay particularly effective. Exponential-based rules for decaying the learning rate were early introduced in literature to speed up the learning process [7, 95]. CL algorithms that exploit this technique [81, 138] do so in a task-wise manner (namely, the schedule starts again at the beginning of each task). Differently, we point out that decreasing the learning rate for the whole duration of the training relieves catastrophic forgetting. We thus recommend computing the learning rate for the j^{th} training example as follows:

$$\text{lr}_j = \text{lr}_0 \cdot \gamma^{N_{\text{ex}}}, \tag{3.4}$$

where N_{ex} is the number of input examples seen so far, lr_0 indicates the initial learning rate for training and γ is a hyper-parameter tuned to make the learning rate approximately $1/6$ of the initial value at the end of the training.

It is worth noting that decreasing the learning rate yields an additional regularisation objective, which penalises weights change between subsequent steps. This achieves a similar effect to the loss terms designed by the parameter-importance family of regularisation-based CL approaches described in Sec. 2.5 [81, 199, 25]. However, these approaches are characterised by extra overhead that is instead avoided by ELrD.

3.3.4 Balanced Reservoir Sampling (BRS)

As outlined in Sec. 3.2, *reservoir* sampling is an online update procedure used to populate a fixed-size buffer with data coming from the input stream. It guarantees each exemplar from that stream to be represented

Alg. 3.1: Balanced Reservoir Sampling

```
1: Input: exemplar (x, y), replay buffer M,
2:        number of seen examples N.
3: if |M| > N then
4:    M[N] ← (x, y)
5: else
6:    j ← RandInt([0, N])
7:    if j < |M| then
                                          Reservoir Sampling
8:       M[j] ← (x, y)
                                          Balanced Reservoir Sampling
9:       ỹ ← argmax ClassCounts(M, y)
10:      k ← RandChoice({k̃; M[k̃] = (x, y), y = ỹ})
11:      M[k] ← (x, y)
12:   end if
13: end if
```

in the buffer with the same probability, making it equivalent to an offline random sampling at each time step. For a dataset featuring C distinct classes and a buffer of size $|\mathcal{M}|$, this implies the following probability of leaving out at least one class:

$$P = \left(1 - \frac{1}{C}\right)^{|\mathcal{M}|}. \tag{3.5}$$

This results in leaving $1/P$ classes out of $\mathcal{M}$, which becomes especially critical when dealing with small buffers: considering $|\mathcal{M}| \approx C$, that probability swiftly increases from 0.25 for $C = 2$ to 0.349 for $C = 10$ (to 0.367 for $C \to \infty$).

Such an issue can be overcome by resorting to the *ring* buffer [28] or the *herding* [138] strategy; however, these are not optimal in terms of buffer exploitation or computational overhead. The former reserves a slice as large as $|\mathcal{M}|/C$ for each class: since classes are shown incrementally, this leaves the main part of the buffer empty. The latter changes the dimension of such slices with the number of seen classes, always reserving $|\mathcal{M}|/C_{\text{seen}}$ slots for each class (where C_{seen} indicates the number of seen classes). Despite its increased efficiency in terms of memory, *herding* additionally requires performing a forward pass over the training set at the end of each task. Instead, we propose Balanced Reservoir Sampling (see Alg. 3.1), which introduces a small modification to *reservoir* to balance the number of exemplars per class within the buffer. Instead of replacing a random exemplar when a newer one is inserted (line 8 of Alg. 3.1),

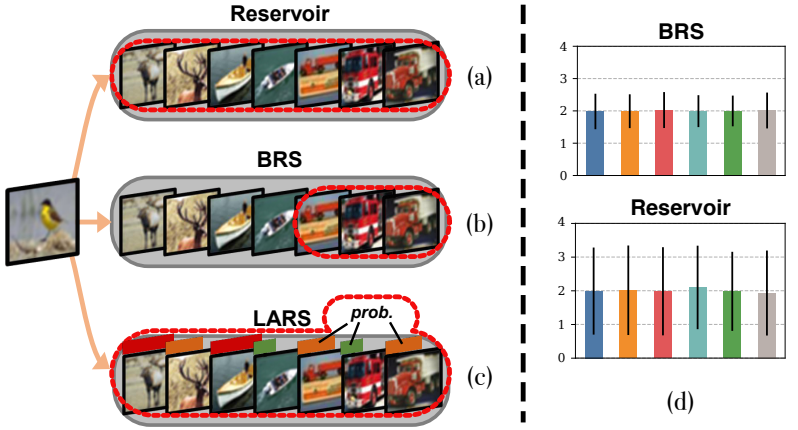


Fig. 3.2: (a-c) comparison of basic, Balanced and Loss-Aware *reservoir*. If a new item is sampled from the stream and $\mathcal{M}$ is full: (a) reservoir randomly discards an exemplar; (b) BRS randomly discards an exemplar from the most represented class; (c) LARS selects what item to discard with a probability given by its loss score. (d) Number of exemplars per class when applying different sampling strategies to the toy dataset described in Sec. 3.3.4, error bars indicate standard deviation. We consider a buffer size of 12 items, with the objective of sampling exactly 2 item per class.

we look for the element to be removed among those belonging to the most represented class (lines 9-11). The difference is also graphically exemplified in Fig. 3.2a-b.

To facilitate understanding, we compare Balanced Reservoir with *reservoir* on a toy dataset of 1020 items belonging to 6 distinct classes (170 items per class). Fig. 3.2d shows the number of samples per class retained at the end of each test. BRS and *reservoir* achieve a Mean Squared Error of 0.28 and 1.64 respectively w.r.t. the ideal solution of storing exactly 2 items per class, giving quantitative evidence for the merit of our proposal.

3.3.5 Loss-Aware Reservoir Sampling (LARS)

To reduce the risk of overfitting buffer data-points, we introduce here an additional variation to the sampling strategy. Taking inspiration from [6], we wish to make room for new examples by discarding those that are less important for preserving the performance. The authors of [6] devise both a rigorous Integer Quadratic Programming-based objective and a more efficient approximated greedy strategy for this purpose. However, since they resort to comparison between the gradients of individual examples, their proposal implies a clear increase in time complexity w.r.t. to plain

Alg. 3.2: Loss-Aware Balanced Reservoir Sampling

Input: exemplar with associated loss score (x, y, s),
replay buffer $\mathcal{M}$, number of seen examples N.
if $|\mathcal{M}| > N$ **then**
  $\mathcal{M}[N] \leftarrow (x, y, s)$
else
  $j \leftarrow \text{RandInt}([0, N])$
  if $j < |\mathcal{M}|$ **then**
    $\boldsymbol{S}_{\text{balance}} \leftarrow \{\text{ClassCounts}(y); \forall (x, y, s) \in \mathcal{M}\}$
    $\boldsymbol{S}_{\text{loss}} \leftarrow \{-s; \forall (x, y, s) \in \mathcal{M}\}$
    $\alpha \leftarrow \sum_k |\boldsymbol{S}_{\text{balance}}[k]| / \sum_k |\boldsymbol{S}_{\boldsymbol{loss}}[k]|$
    $\boldsymbol{S} \leftarrow \boldsymbol{S}_{\text{loss}} \cdot \alpha + \boldsymbol{S}_{\text{balance}}$
    $\boldsymbol{probs} \leftarrow \boldsymbol{S} / \sum_k \boldsymbol{S}[k]$
    $k \leftarrow \text{RandInt}([0, |\mathcal{M}|], \boldsymbol{probs})$
    $\mathcal{M}[k] \leftarrow (x, y, s)$
  end if
end if

reservoir. Instead, we propose using the training loss value directly as a much simpler yet effective indicator of example importance. Indeed, the overall expected loss of the buffer can be computed without back-propagation and it should be maximised at all times, thus promoting the retention of exemplars that have not been fit (see Fig. 3.2c).

Making *reservoir loss-aware*, requires identifying and replacing the elements displaying low loss values. These can be naïvely computed by feeding all the replay examples into the model before the replacement phase. However, this becomes computationally inefficient when the buffer is large, motivating us to adopt an online update of the loss values: for every example stored in the buffer, we also save the original loss score incurred in its optimisation. As this is a scalar value, the memory overhead that results from storing it is negligible w.r.t. the cost of storing the example to be replayed. To keep the scores up to date, whenever the corresponding items are drawn for replay we replace the stored loss values with the current losses that are yielded by the model.

Since they are complementary and address separate issues, we combine Loss-Aware Reservoir (LARS) and BRS into a single algorithm (Alg. 3.2). To do so, we: *i)* compute a S_{balance} score vector given by the frequency of each class (line 8); *ii)* estimate an importance score S_{loss}, given by the negative loss value of each example (line 9); *iii)* normalise these two terms to ensure an equal contribution and sum them to form a single score vector S (lines 10-11). Finally, we assign a replacement probability to each item proportional to the combined score (lines 12-14).

FAA	**S-FMNIST**			**S-CIF10**		
FT	20.11			19.62		
JT	84.47			92.13		
$\|\mathcal{M}\|$	200	500	1000	200	500	1000
A-GEM	49.73	49.47	50.98	19.90	20.35	19.81
GEM	69.46	75.91	79.62	28.14	34.69	36.68
HAL	72.59	77.59	80.79	25.92	27.99	29.10
iCaRL	75.46	77.54	78.13	41.26	41.34	42.03
ER	72.54	79.02	81.39	24.06	27.06	31.38
ER+Tricks	**76.07**	**80.11**	**82.46**	**59.18**	**62.60**	**70.99**

Tab. 3.1: Class-IL FAA of SOTA RBMs on S-FMNIST and S-CIF10.

FAA	**S-CIF100**			**S-CORe50**		
FT	8.54			8.89		
JT	70.66			49.51		
$\|\mathcal{M}\|$	200	500	1000	200	500	1000
A-GEM	9.17	9.23	9.12	9.33	9.42	8.96
GEM	9.18	14.12	17.88	–	–	–
HAL	7.63	9.66	10.43	11.53	12.40	8.53
iCaRL	20.73	24.74	25.52	8.01	7.23	8.05
ER	9.66	11.50	12.36	19.48	28.54	32.66
ER+Tricks	**21.26**	**24.90**	**36.05**	**25.63**	**33.33**	**37.44**

Tab. 3.2: Class-IL FAA of SOTA RBMs on S-CIF100 and S-CORe50.

3.4 Experiments

We test our proposal on four Class-IL settings, following the specifics outlined in Sec. 2.3: S-FMNIST, S-CIF10, S-CIF100[4] and S-CORe50. Results are expressed as FAA and averaged over 10 independent runs.

3.4.1 Comparison with the State of the Art

In this section, we draw a comparison between ER equipped with our tricks (ER+Tricks) and SOTA RBMs, namely iCaRL, GEM, A-GEM and HAL. We test ER in combination with the *reservoir* sampling strategy, while GEM, A-GEM, HAL use a *ring* buffer and iCaRL employs *herding*. Tab. 3.1 and 3.2 report the results for different buffer sizes (200, 500, 1000), with FT and JT respectively providing a lower and upper bound.

The experiments on S-FMNIST show that ER+Tricks consistently surpasses all competitors. Due to the mentioned weakness of *reservoir*, the

[4]For S-CIF100 we deviate from Tab. 2.1 and apply no learning rate decay.

FAA	S-FMNIST	S-CIF10	S-CIF100
ER	72.54	24.06	9.66
+ IBA	–	44.78	13.90
+ BiC	73.43	49.27	17.73
+ ElrD	74.19	51.02	20.27
+ BRS	74.66	52.75	20.64
+ LARS	**76.07**	**59.18**	**21.26**

Tab. 3.3: Class-IL FAA of ER as more tricks from Sec 3.3 are added ($|\mathcal{M}| = 200$).

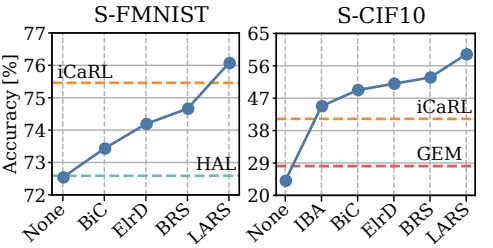


Fig. 3.3: FAA on S-FMNIST and S-CIF10 as more training tricks are applied to ER, with $|\mathcal{M}| = 200$. FAA of competitor methods also reported.

tricks prove especially beneficial when the memory buffer is smaller. However, ER proves to be an already strong baseline, as revealed by GEM and A-GEM consistently performing worse than it. HAL starts out on par with ER at reduced buffer size but achieves weaker results when the latter increases. Interestingly, iCaRL only performs better than ER at memory size 200. This is due to the *herding* strategy, through which the method fills the buffer with the best possible exemplars for its classification procedure. By so doing, iCaRL gains an advantage over methods using *reservoir* and *ring* sampling that is however less prominent for larger $|\mathcal{M}|$.

Experiments on the harder S-CIF10 and S-CIF100 protocols show GEM and iCaRL prevailing over naïve ER. In particular, iCaRL sets a very high bar on hard datasets thanks to its simple and effective nearest-mean-of-exemplars classification rule. Nevertheless, the application of our proposed tricks gives ER+Tricks an edge over the competition. Notably, HAL encounters some failures on S-CIF100.

Even on S-CORe50, the tricks yield a boost in performance compared to naïve replay, outperforming the other approaches. In contrast with what we have observed for S-CIF10 and S-CIF100, iCaRL does not achieve reliable performance. We ascribe this finding to the nature of S-CORe50: this dataset shows very similar entities (*e.g.*, slightly different plug adapters) as separate classes. While such subtle differences are successfully learnt by ER, iCaRL strictly depends on its nearest-mean-of-exemplars classifier, which struggles to distinguish such fine-grained details.

3.4.2 Influence of Each Trick

To quantify the effect of each trick presented in Sec. 3.3, we apply them increasingly to ER (Tab. 3.3, Fig. 3.3). Since we do not apply data augmentation to the input batches of S-FMNIST, we do not employ it on buffer points either. IBA proves to be very effective on both S-CIF10 and

FAA	S-CIF10			S-CIF100		
$\|\mathcal{M}\|$	200	500	1000	200	500	1000
A-GEM	19.90	**20.35**	19.81	**9.17**	9.23	9.12
+ IBA	**20.23**	19.97	**21.15**	9.16	**9.34**	**9.39**
GEM	**28.14**	**34.69**	**36.68**	9.18	14.12	**17.88**
+ IBA	22.62	23.01	20.25	**13.69**	**16.74**	15.21
HAL	25.92	27.99	29.10	7.63	9.66	10.43
+ IBA	**32.33**	**41.77**	**49.28**	**8.19**	**11.39**	**12.91**

Tab. 3.4: Evaluation of the impact of IBA on Class-IL (Sec. 3.3.1).

FAA	No tricks	BiC	CBiC	CBiC+ElrD
SI	19.91	24.67	33.15	**35.51**
oEWC	20.04	25.71	40.36	**43.85**

Tab. 3.5: Class-IL FAA of regularisation methods on S-FMNIST when adding a buffer with $|\mathcal{M}| = 500$.

S-CIF100, providing a very meaningful FAA boost when compared to the initial performance. This turns out to be especially remarkable in the former setting, where it almost doubles the initial accuracy. Both BiC and ELrD present a solid positive effect, especially when considering the two most difficult settings. Finally, the proposed sampling strategies (BRS and LARS) prove particularly beneficial on S-FMNIST and S-CIF10.

Overall, we observe a remarkable performance boost on S-CIF10 and S-CIF100 (+146% and +120% respectively), showing that the proposed tricks are very effective on challenging datasets where ER leaves room for improvement.

3.4.3 Applicability of IBA to Other RBMs

As it allows to virtually draw from a larger amount of data and often entails no additional costs in terms of annotations or storage, one would expect data augmentation to always be beneficial for the training of RBMs. Surprisingly, by testing the proposed IBA on competitor RBMs, we find that this is not always the case. Interestingly, Tab. 3.4 shows that there is not a clear trend: while HAL always improves its performance in a consistent way, GEM suffers from a severe degradation on S-CIF10. We conjecture that this could be due to the way GEM makes use of the buffer: the inequality constraints given by augmented examples prove sub-optimal for retaining the performance on the original tasks.

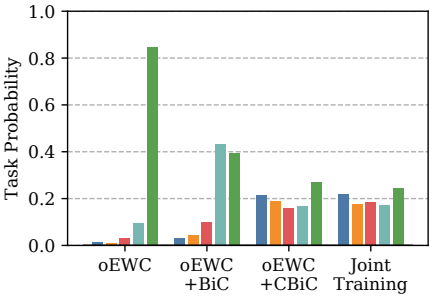


Fig. 3.4: Prediction probability for each task at the end of S-FMNIST, for oEWC with different bias correction layers and JT.

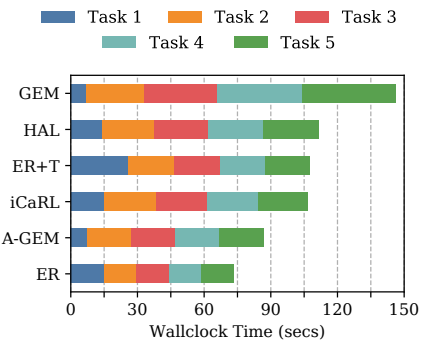


Fig. 3.5: Run times of CL methods on S-FMNIST with $|\mathcal{M}| = 200$.

3.4.4 Applicability to non-Rehearsal Methods

In this section, we showcase the application of BiC and ElrD to two prior-based approaches, namely oEWC [149] and SI [199]. We recall from Sec. 2.5 that these operate by identifying the most important parameters whenever a task change occurs, subsequently preventing their change. While effective in the Task-IL scenario, they tend to completely forget all tasks but the latest in Class-IL. As observed in [182], this is mainly due to an implicit bias induced by the optimisation of the current task.

As BiC only compensates the responses related to the last task, it proves effective only under the hypothesis that previous tasks are equally biased. This holds true for ER and other RBMs, which rehearse an equal number of exemplars from all previous tasks at each iteration. Conversely, oEWC and SI do not access past examples at all, which discourages the classifier from predicting older tasks. To shed light on these issues, we conduct the following simple experiment: *i)* for each test example, we compute the output distribution over the classes; *ii)* we average the probabilities over all exemplars task-wise; *iii)* we normalise the results to describe them as probability distributions and show them in Fig. 3.4. We observe that BiC effectively reduces the bias displayed by oEWC towards the last task. However, since the model also shows a tendency to predict classes of earlier tasks, BiC cannot manage to balance out all prior probabilities.

For this reason, we combine oEWC and SI with a tailored variant of BiC, which we call **Complete Bias Correction (CBiC)**. Whilst BiC only corrects the logits related to the latest task, CBiC adjusts the responses related to each task independently. Technically, CBiC applies an additive offset β_t to the logits related to each task t. Fig. 3.4 shows that this approach favourably results in a flatter distribution for oEWC, which is closer to the

one of JT. This is confirmed by the FAA results reported in Tab. 3.5, which also illustrates how also applying ElrD further increases its performance.

As a final note, we remark that ER is evenly biased w.r.t. previous tasks, making BiC and CBiC equally effective on it. For this reason, we advise using the former for ER as it is simpler and thus easier to train.

For a fair comparison among the listed methods, we report their run-times in Fig. 3.5. This experiment employs the S-FMNIST dataset, with $|\mathcal{M}| = 200$ and identical run conditions for all methods (a single Nvidia GeForce RTX 2080).

Thanks to its simplicity, plain ER is remarkably faster than all other methods. Conversely, GEM is by far the slowest method: as it relies on a demanding QP constraint for each task, its wall-clock time increases remarkably as the training progresses. By comparison, A-GEM is clearly much faster, given that it only applies one similar constraint at all times. iCaRL, HAL and ER+Tricks have similar medium-to-long execution times linked to the extra computation required at task boundaries. In this phase, the *herding* strategy of iCaRL examines all examples of the previous task, HAL computes one anchor point per class and ER+Tricks trains the Bias Control module. A potentially slowing-down factor for our proposal is the increased amount of computation required by BRS and LARS; however, as *reservoir* samples more frequently at the beginning of training, this effect fades in subsequent tasks.

3.5 Conclusions

In this chapter, we introduced a collection of simple training tricks meant to enhance ER on the Class-IL setting. The effectiveness of our proposals was shown by means of an experimental comparison among SOTA RBMs on increasingly harder datasets: at the cost of a limited growth in computational requirements, ER equipped with our tricks outperforms more sophisticated approaches. Finally, we showed that some of the tricks can be beneficially applied even to regularisation CL methods.

In proposing novel RBMs, the following chapters will consistently apply IBA, shown here as one of the most effective training tricks, while Chap. 7 will leverage BRS as it explicitly requires a balanced memory buffer for the design of geometrical learning constraints. Our next proposals do not apply the other tricks presented in this chapter; however, several of them will provide alternative approaches for dealing with the biased prediction problem addressed by BiC.

Following the original proposal of these tricks, other influential works have studied the same topics. Among them, we find approaches designed to remove prediction bias [3, 19], a detailed investigation into the role of factors such as learning rate decay in continual learning [113], balanced variants of *reservoir* sampling very similar to our BRS [78, 33].

Chapter 4

Knowledge Distillation Replay along the Training Trajectory

4.1 Motivation

In the previous chapter, we have shown that ER constitutes a very reliable baseline for CL thanks to its simple formulation. Here, we propose an improved baseline method that pursues improved performance while remaining in line with the simplicity of ER. Our proposal stems from the observation that the replay objective of ER is entirely *model-independent*: as every replay example is presented to the learner in pair with its ground-truth target, the learning signal does not depend on the learner at all: it is merely a repeated presentation of the same raw information that was originally made available on the input stream.

Here, we propose a change in the replay objective, breaking the symmetry with current-task data and requiring that the model learn from past data through self-knowledge distillation [59]. Our resulting approach is called **Dark Experience Replay (DER)** because the new learning objective facilitates the transfer of *dark knowledge* [58], *i.e.*, a richer characterisation of the replay examples conveyed by the unabridged probability distribution of past model responses.

In contrast with other CL approaches that resort to knowledge distillation [96, 138], our proposal is less memory-intensive, since it does not require the availability of a previous model snapshot, but rather stores the distillation targets directly in the memory buffer. Furthermore, our approach samples its distillation targets throughout the entire optimisation trajectory, rather than learning from a fully converged teacher (*e.g.*, at the previous task boundary). While this may seem sub-optimal, we empirically show that our baseline exhibits remarkable qualities: it converges to flatter minima and achieves better model calibration at the

cost of a limited memory and training time overhead.

The design of DER aims at a fully online operation with no reliance on task boundaries at any time of the CL training and easy adaptability to both sudden and gradual changes in the input distribution. This makes our approach naturally compatible with the GCL setting that will be presented later in Chap. 8.

4.2 Dark Experience Replay

Designing a Class-IL CL model implies devising an opportune $\mathcal{L}_{\mathrm{R}}$ for Eq. 2.2 with the purpose of allowing the model to fit the current task well while still faithfully approximating the behaviour observed on the old ones. An effective approach towards this goal which does not over-constrain the network's parameters is given by simply requiring the learner to mimic its original responses for past samples, in a typical self-distillation paradigm:

$$\mathcal{L}_{\mathrm{R}} = \alpha \sum_{i=0}^{c-1} \mathop{\mathbb{E}}_{x \sim \mathcal{T}_i} \Big[D_{\mathrm{KL}} \big(f_{\theta_i^*}(x) \;||\; f_\theta(x) \big) \Big], \tag{4.1}$$

where θ_i^* denotes the optimal set of parameters at the end of task i, $\mathcal{T}_c$ is the current task and α is a hyper-parameter meant to balance the trade-off between the plasticity and stability of the CL model in Eq. 2.2. In spite of its simplicity, the objective in Eq. 4.1 requires the availability of $\mathcal{T}_i$ data for previous tasks, which violates the constraints of CL. To compensate for the absence of this data source, we need to introduce a replay buffer $\mathcal{M}_t$ dedicated to storing past *experiences* for task t. Differently from typical RBMs [6, 26, 141], we do not use the buffer for storing the ground-truth labels y, but rather the network's logits $\ell \triangleq h_{\theta_t}(x)$[1]. We therefore rewrite Eq. 4.1 as follows:

$$\mathcal{L}_{\mathrm{R}} = \alpha \sum_{t=0}^{c-1} \mathop{\mathbb{E}}_{(x,\ell) \sim \mathcal{M}_t} \Big[D_{\mathrm{KL}} \big(\mathrm{softmax}(\ell) \;||\; f_\theta(x) \big) \Big]. \tag{4.2}$$

To lift the dependency on known task boundaries for buffer population, we adopt *reservoir* sampling [173]. By so doing, we keep a unified memory buffer $\mathcal{M}$, guaranteeing that all examples on the input stream are given the same probability of being stored in it. This motivates a further rewrite of Eq. 4.2 as follows:

$$\mathcal{L}_{\mathrm{R}} = \alpha \cdot \mathop{\mathbb{E}}_{(x,\ell) \sim \mathcal{M}} \Big[D_{\mathrm{KL}} \big(\mathrm{softmax}(\ell) \;||\; f_\theta(x) \big) \Big]. \tag{4.3}$$

It should be noted that the strategy outlined in Eq. 4.3 implies sampling logits ℓ along the optimisation trajectory. These targets may account

[1] With h_θ indicating pre-softmax model responses of $f_\theta(\cdot)$, as defined in Sec. 3.3.2.

for a backbone behaviour that diverges from the one of a teacher fully converged on the reference task. Even if this is counter-intuitive, we empirically observe that the usage of such *sub-optimal* logits does not hurt performance and that it produces beneficial effects in terms of flatness of the attained minima and calibration (see Sec. 4.4).

Under mild assumptions [59], the optimisation of the KL divergence in Eq. 4.3 is equivalent to minimising the Euclidean distance between the corresponding pre-softmax responses (*i.e.*, logits); such a formulation should be preferred as it avoids the information loss occurring in probability space due to the squashing function (*e.g.*, softmax) [102]. This allows us to provide the final formulation for our proposed **Dark Experience Replay (DER)**, which we also outline in Alg. 4.1:

$$\mathcal{L}_{\text{DER}} \triangleq \alpha \cdot \mathop{\mathbb{E}}_{(x,\ell)\sim\mathcal{M}} \Big[\|\ell - h_\theta(x)\|_2^2 \Big]. \tag{4.4}$$

4.2.1 Dark Experience Replay++

On the one hand, *reservoir* allows DER not to depend on known task boundaries; on the other, it might introduce a vulnerability in the case of sudden distribution shifts occurring in the input stream. In this case, logits that are highly biased by the training on previous tasks might be sampled for later replay, providing unreliable information. Such a shortcoming can be mitigated by replaying ground-truth labels – as done by ER – in conjunction with logits. For this reason, we further propose **Dark Experience Replay++ (DER++)**, which combines the objective of Eq. 4.4 with an additional term promoting higher conditional likelihood of buffer data-points w.r.t. their ground-truth labels:

$$\mathcal{L}_{\text{DER++}} \triangleq \alpha \mathop{\mathbb{E}}_{(x,y,\ell)\sim\mathcal{M}} \Big[\|\ell - h_\theta(x)\|_2^2 \Big] \; + \; \beta \mathop{\mathbb{E}}_{(x',y',\ell')\sim\mathcal{M}} \Big[\text{CE}\big(y', f_\theta(x')\big) \Big], \tag{4.5}$$

where β is an additional coefficient balancing the last term[2] (DER++ collapses to DER when $\beta = 0$). The extended objective of Eq. 4.5 produces a minimal overhead w.r.t. Eq. 4.4; the detailed procedure for DER++ can be found in Alg. 4.2.

4.2.2 Relation with Other Distillation-Based CL Methods

While both our proposal and LwF [96] leverage self-knowledge distillation in CL, they adopt remarkably different approaches. LwF does not replay past examples, but rather adopts the model at the last task boundary as a teacher and encourages the similarity between its responses and the ones of the current learner w.r.t. current-task data. Alternatively, iCaRL [138]

[2]The model is not overly sensitive to α and β: setting them both to 0.5 yields stable performance.

Alg. 4.1: Dark Experience Replay

Input: dataset D, parameters θ, scalar α, learning rate λ
$\mathcal{M} \leftarrow \{\}$
for (x, y) **in** D **do**
  $(x', \ell') \leftarrow \text{sample}(\mathcal{M})$
  $(x_t, x'_t) \leftarrow (\text{augment}(x), \text{augment}(x'))$
  $\ell \leftarrow h_\theta(x_t)$
  $\theta \leftarrow \theta + \lambda \nabla_\theta [\text{CE}(y, f_\theta(x_t)) + \alpha \left\| \ell - h_\theta(x'_t) \right\|_2^2$
  $\mathcal{M} \leftarrow \text{reservoir}(\mathcal{M}, (x, \ell))$
end for

Alg. 4.2: Dark Experience Replay++

Input: dataset D, parameters θ, scalars α, β, learning rate λ
$\mathcal{M} \leftarrow \{\}$
for (x, y) **in** D **do**
  $(x', y', \ell') \leftarrow \text{sample}(\mathcal{M})$
  $(x'', y'', \ell'') \leftarrow \text{sample}(\mathcal{M})$
  $(x_t, x'_t, x''_t) \leftarrow (\text{augment}(x), \text{augment}(x'), \text{augment}(x''))$
  $\ell \leftarrow h_\theta(x_t)$
  $\theta \leftarrow \theta + \lambda \nabla_\theta [\text{CE}(y, f_\theta(x_t)) + \alpha \left\| \ell - h_\theta(x'_t) \right\|_2^2 + \beta \, \text{CE}(y'', f_\theta(x''_t))]$
  $\mathcal{M} \leftarrow \text{reservoir}(\mathcal{M}, (x, y, \ell))$
end for

distils knowledge for past outputs w.r.t. past exemplars, which is similar to our proposed methods. However, the former exploits the network appointed at the end of each task as the sole teaching signal. On the contrary, our methods store logits sampled throughout the optimisation trajectory, which could be compared to having a slightly different teacher for each replay example.

A proposal which is very similar in spirit to DER is given by the application of Function Distance Regularisation (FDR) for catastrophic forgetting prevention (Sec. 3.1 of [15]). Like FDR, we use past exemplars and network outputs to align past and current outputs. However – similarly to iCaRL – FDR depends on the availability of task boundaries for storing network responses at convergence. The empirical analysis we present in Sec. 4.4 indicates that not only this requirement can be relaxed without experiencing a drop in performance, but doing so endows DER and DER++ with additional remarkable properties and allows them to attain higher FAA on the evaluated benchmarks. All things considered, we remark that the motivation behind [15] lies chiefly in studying how the training trajectory of DNNs can be characterised in a functional L^2 Hilbert space, whereas the potential of function-space regularisation for

Continual Learning problems is only coarsely addressed with a single experiment on S-MNIST. In this respect, we present extensive experiments on multiple CL settings as well as a detailed analysis (Sec. 4.4) providing a deeper understanding on the effectiveness of this kind of regularisation.

4.3 Experiments

We present here an extensive suite of CL benchmarks encompassing all the *academic* settings introduced in Sec. 2.2.1. Experiments on the Class-IL and Task-IL protocols encompass S-MNIST, S-CIF10 and S-TinyImg; experiments on the Domain-IL setting exploit P-MNIST and R-MNIST. Results are presented in terms of FAA, FAF, FBWD and FFWD, averaged over 10 independent runs.

Our evaluation presents the following competitors: two regularisation-based methods (oEWC, SI), two methods leveraging Knowledge Distillation (iCaRL, LwF), one architectural method (PNN) and seven RBMs (ER, GEM, A-GEM, GSS, FDR, HAL, MER[3]). We report the results of our evaluations in Tab. 4.1-4.4 for FAA, Tab. 4.5 for FAF, Tab. 4.6 for FFWD and Tab. 4.7 for FBWD; in these tables, '-' indicates experiments that could not be run, either due to setting compatibility issues (*e.g.*, PNN, iCaRL and LwF on Domain-IL) or intractable training time (*e.g.*, GEM, HAL and GSS on S-TinyImg).

DER and DER++ achieve SOTA performance in almost all presented settings. When compared to oEWC and SI, the gap in FAA across all settings appears unbridgeable, suggesting that regularisation towards old sets of parameters does not suffice to prevent forgetting. We argue that this could also be due to weights importance being computed in earlier tasks and thus failing to adapt to subsequent training phases. While being computationally more efficient, LwF performs worse than SI and oEWC on average. PNN, which achieves the strongest results among non-rehearsal methods, attains lower accuracy than replay-based ones in spite of its memory footprint being much higher at any buffer size.

When compared to rehearsal methods, DER and DER++ show strong performance in the majority of benchmarks, especially in the Domain-IL scenario (Tab. 4.4). For these problems, a shift occurs on the input domain, but not on classes: hence, the relations among them also likely persist[4]. For this reason, we argue that leveraging soft targets in place of hard ones (ER) carries valuable information [59], exploited by DER and DER++ to preserve the inter-class similarity structures through the data-stream.

[3]MER is only presented on S-MNIST as we experienced an intractable training time on other benchmarks (*e.g.*, while DER takes approximately 2.5 hours on S-CIF10, MER takes 300 hours).

[4]As an example, if it is true that during the first task of R-MNIST number 2s visually look like 3s, this still holds true when a different rotation is applied in the following tasks.

FAA	S-MNIST					
Method	**Class-IL**			**Task-IL**		
JT	95.57±0.24			99.51±0.07		
FT	19.60±0.04			94.94±2.18		
oEWC	**20.46±1.01**			98.39±0.48		
SI	19.27±0.30			96.00±2.04		
LwF	19.62±0.01			94.11±3.01		
PNN	-			**99.23±0.20**		
$\|\mathcal{M}\|$	200	500	5120	200	500	5120
ER	80.43±1.89	86.12±1.89	93.40±1.29	97.86±0.35	**99.04±0.18**	99.33±0.22
MER	81.47±1.56	88.35±0.41	94.57±0.18	98.05±0.25	98.43±0.11	99.27±0.09
GEM	80.11±1.54	85.99±1.35	95.11±0.87	97.78±0.25	98.71±0.20	99.44±0.12
A-GEM	45.72±4.26	46.66±5.85	54.24±6.49	98.61±0.24	98.93±0.21	98.93±0.20
iCaRL	70.51±0.53	70.10±1.08	70.60±1.03	98.28±0.09	98.32±0.07	98.32±0.11
FDR	79.43±3.26	85.87±4.04	87.47±3.15	97.66±0.18	97.54±1.90	97.79±1.33
GSS	38.90±2.49	49.76±4.73	89.39±0.75	95.02±1.85	97.71±0.53	98.33±0.17
HAL	84.70±0.87	87.21±0.49	89.52±0.96	97.96±0.21	98.03±0.22	98.35±0.17
DER	84.55±1.64	90.54±1.18	94.90±0.57	**98.80±0.15**	98.84±0.13	99.29±0.11
DER++	**85.61±1.40**	**91.00±1.49**	**95.30±1.20**	98.76±0.28	98.94±0.27	**99.47±0.07**

Tab. 4.1: FAA results on S-MNIST.

FAA	S-CIF10					
Method	**Class-IL**			**Task-IL**		
JT	92.20±0.15			98.31±0.12		
FT	19.62±0.05			61.02±3.33		
oEWC	**19.49±0.12**			68.29±3.92		
SI	19.48±0.17			68.05±5.91		
LwF	19.46±0.31			63.65±1.80		
PNN	-			**95.13±0.72**		
$\|\mathcal{M}\|$	200	500	5120	200	500	5120
ER	48.33±1.57	60.98±1.48	84.30±0.73	91.49±0.92	**94.19±0.32**	**97.02±0.15**
GEM	25.54±0.76	26.20±1.26	25.26±3.46	90.44±0.94	92.16±0.69	95.55±0.02
A-GEM	20.04±0.34	22.67±0.57	21.99±2.29	83.88±1.49	89.48±1.45	90.10±2.09
iCaRL	60.58±1.32	55.42±4.16	63.47±1.33	**93.97±0.53**	91.43±1.84	95.47±0.26
FDR	30.91±2.74	28.71±3.23	19.70±0.07	91.01±0.68	93.29±0.59	94.32±0.97
GSS	39.07±5.59	49.73±4.78	67.27±4.27	88.80±2.89	91.02±1.57	94.19±1.15
HAL	34.90±2.55	46.19±4.14	64.99±3.71	83.14±3.66	86.08±2.48	89.01±2.64
DER	61.93±1.79	70.51±1.67	83.81±0.33	91.40±0.92	93.40±0.39	95.43±0.33
DER++	**64.88±1.17**	**72.70±1.36**	**85.24±0.49**	91.92±0.60	93.88±0.50	96.12±0.21

Tab. 4.2: FAA results on S-CIF10.

FAA			**S-TinyImg**			
Method		**Class-IL**			**Task-IL**	
JT		59.99±0.19			82.04±0.10	
FT		7.92±0.26			18.31±0.68	
oEWC		7.58±0.10			19.20±0.31	
SI		6.58±0.31			36.32±0.13	
LwF		**8.57±0.11**			16.57±0.37	
PNN		-			**67.84±0.29**	
$\|\mathcal{M}\|$	200	500	5120	200	500	5120
ER	8.77±0.17	11.06±0.32	29.93±0.47	38.97±0.97	49.89±0.73	67.89±0.50
A-GEM	8.07±0.08	8.06±0.04	7.96±0.13	22.77±0.03	25.33±0.49	26.22±0.65
iCaRL	**14.72±0.59**	**20.18±0.56**	31.60±0.33	**42.84±0.92**	**52.07±0.58**	64.54±0.30
FDR	8.70±0.19	10.54±0.21	28.97±0.41	40.36±0.68	49.88±0.71	68.01±0.42
DER	11.87±0.78	17.75±1.14	36.73±0.64	40.22±0.67	51.78±0.88	69.50±0.26
DER++	10.96±1.17	19.38±1.41	**39.02±0.97**	40.87±1.16	51.91±0.68	**69.84±0.63**

Tab. 4.3: FAA results on S-TinyImg.

FAA				**Domain-IL**		
Method		**P-MNIST**			**R-MNIST**	
JT		94.33±0.17			95.76±0.04	
FT		40.70±2.33			67.66±8.53	
oEWC		**75.79±2.25**			**77.35±5.77**	
SI		65.86±1.57			71.91±5.83	
$\|\mathcal{M}\|$	200	500	5120	200	500	5120
ER	72.37±0.87	80.60±0.86	89.90±0.13	85.01±1.90	88.91±1.44	93.45±0.56
GEM	66.93±1.25	76.88±0.52	87.42±0.95	80.80±1.15	81.15±1.98	88.57±0.40
A-GEM	66.42±4.00	67.56±1.28	73.32±1.12	81.91±0.76	80.31±6.29	80.18±5.52
FDR	74.77±0.83	83.18±0.53	90.87±0.16	85.22±3.35	89.67±1.63	94.19±0.44
GSS	63.72±0.70	76.00±0.87	82.22±1.14	79.50±0.41	81.58±0.58	85.24±0.59
HAL	74.15±1.65	80.13±0.49	89.20±0.14	84.02±0.98	85.00±0.96	91.17±0.31
DER	81.74±1.07	87.29±0.46	91.66±0.11	90.04±2.61	92.24±1.12	94.14±0.31
DER++	**83.58±0.59**	**88.21±0.39**	**92.26±0.17**	**90.43±1.87**	**92.77±1.05**	**94.65±0.33**

Tab. 4.4: FAA results on P-MNIST and R-MNIST.

		FAF					
$\|\mathcal{M}\|$	Method	S-MNIST		S-CIF10		P-MNIST	R-MNIST
		Class-IL	Task-IL	Class-IL	Task-IL	Domain-IL	Domain-IL
–	FT	99.10±0.55	5.15±2.74	96.39±0.12	46.24±2.12	57.65±4.32	20.82±2.47
–	oEWC	**97.79±1.24**	0.44±0.16	**91.64±3.07**	29.33±3.84	36.69±2.34	36.44±1.44
	SI	98.89±0.86	5.15±2.74	95.78±0.64	38.76±0.89	**27.91±0.31**	**23.41±0.49**
	LwF	99.30±0.11	5.15±2.74	96.69±0.25	32.56±0.56	-	-
	PNN	-	**0.00±0.00**	-	**0.00±0.00**	-	-
200	ER	21.36±2.46	0.84±0.41	61.24±2.62	7.08±0.64	22.54±0.95	8.87±1.44
	MER	20.38±1.97	0.82±0.21	-	-	-	-
	GEM	22.32±2.04	1.19±0.38	82.61±1.60	9.27±2.07	29.38±2.56	12.97±4.82
	A-GEM	66.15±6.84	0.96±0.28	95.73±0.20	16.39±0.86	31.69±3.92	20.05±1.12
	iCaRL	**11.73±0.73**	**0.28±0.08**	**28.72±0.49**	**2.63±3.48**	-	-
	FDR	21.15±4.18	0.52±0.18	86.40±2.67	7.36±0.03	20.62±0.65	13.66±2.52
	GSS	74.10±3.03	4.30±2.31	75.25±4.07	8.56±1.78	47.85±1.82	20.71±6.50
	HAL	14.54±1.49	0.53±0.19	69.11±4.21	12.26±0.02	79.00±1.17	83.59±0.04
	DER	17.66±2.10	0.57±0.18	40.76±0.42	6.57±0.20	14.00±0.73	6.53±0.32
	DER++	16.27±1.73	0.66±0.28	32.59±2.32	5.16±0.21	**11.49±0.31**	**6.08±0.43**
500	ER	15.97±2.46	0.39±0.20	45.35±0.07	3.54±0.35	14.90±0.39	8.02±1.56
	MER	11.52±0.56	0.45±0.17	-	-	-	-
	GEM	15.57±1.77	0.54±0.15	74.31±4.62	9.12±0.21	18.76±0.91	8.79±1.44
	A-GEM	65.84±7.24	0.64±0.20	94.01±1.16	14.26±4.18	28.53±2.01	19.70±3.14
	iCaRL	11.84±0.73	**0.30±0.09**	25.71±1.10	**2.66±2.47**	-	-
	FDR	13.90±5.19	1.35±2.40	85.62±0.36	4.80±0.00	12.80±1.28	7.21±1.89
	GSS	60.35±6.03	0.89±0.40	62.88±2.67	7.73±3.99	23.68±1.35	18.05±9.89
	HAL	9.97±1.62	0.35±0.21	62.21±4.34	5.41±1.10	82.53±0.36	88.53±0.77
	DER	9.58±1.52	0.45±0.13	26.74±0.15	4.56±0.45	8.07±0.43	3.96±2.08
	DER++	**8.85±1.86**	0.35±0.15	**22.38±4.41**	4.66±1.15	**7.67±1.05**	**3.57±0.09**
5120	ER	6.08±1.84	0.25±0.23	13.99±1.12	0.27±0.06	5.24±0.13	3.10±0.42
	MER	**3.22±0.33**	**0.07±0.06**	-	-	-	-
	GEM	4.30±1.16	0.16±0.09	75.27±4.41	6.91±2.33	6.74±0.49	2.49±0.17
	A-GEM	55.10±10.79	0.63±0.21	84.49±3.08	11.36±1.68	23.74±2.23	18.10±1.44
	iCaRL	11.64±0.72	0.26±0.06	24.94±0.14	1.59±0.57	-	-
	FDR	11.58±3.97	0.95±1.61	96.64±0.19	1.93±0.48	3.82±0.12	3.31±0.56
	GSS	7.90±1.21	0.18±0.11	58.11±9.12	7.71±2.31	89.76±0.39	92.66±0.02
	HAL	6.55±1.63	0.13±0.07	27.19±7.53	5.21±0.50	19.97±1.33	17.62±2.33
	DER	4.53±0.83	0.32±0.08	10.12±0.80	2.59±0.08	3.51±0.03	2.17±0.11
	DER++	4.19±1.63	0.23±0.06	**7.27±0.84**	**1.18±0.19**	**2.96±0.14**	**1.62±0.50**

Tab. 4.5: FAF results for the Experiments of Sec. 4.3.

		FFWD					
$\|\mathcal{M}\|$	Method	S-MNIST		S-CIF10		P-MNIST	R-MNIST
		Class-IL	Task-IL	Class-IL	Task-IL	Domain-IL	Domain-IL
–	FT	−11.06±2.90	2.33±4.71	−9.09±0.11	−1.46±1.17	0.32±0.85	48.94±0.10
–	oEWC	**−7.44±4.18**	**−0.13±8.12**	−12.51±0.02	−4.09±7.97	0.69±0.97	52.45±8.75
	SI	−9.50±5.27	−1.34±5.42	−12.64±0.20	−2.33±2.29	**0.71±1.89**	**53.09±0.73**
	LwF	−12.39±4.06	1.30±5.40	**−10.63±5.12**	**0.73±4.36**	-	-
	PNN	-	N/A	-	N/A	-	-
200	ER	−12.12±2.21	−0.86±3.24	−11.02±2.77	2.10±1.27	1.37±0.48	66.79±0.05
	MER	−11.03±3.40	−2.18±3.51	-	-	-	-
	GEM	−10.26±3.08	−0.16±5.89	−7.50±7.05	0.13±3.54	0.42±0.35	54.06±4.35
	A-GEM	**−10.04±3.11**	2.39±6.96	−11.37±0.08	−0.34±0.13	0.83±0.57	54.84±10.45
	iCaRL	N/A	N/A	N/A	N/A	-	-
	FDR	−12.06±2.22	−0.81±3.89	−12.75±0.30	−2.42±0.86	−1.24±0.06	60.71±8.17
	GSS	−11.31±2.58	2.99±6.61	−7.08±10.01	**6.17±2.06**	0.04±0.85	57.28±4.47
	HAL	−11.15±3.56	−0.20±3.99	−11.94±0.80	−0.02±0.10	**1.72±0.08**	59.95±3.71
	DER	−10.16±3.78	**3.23±5.24**	−11.89±0.88	0.27±7.12	1.23±0.26	64.69±2.02
	DER++	−12.42±1.84	−2.33±5.69	**−4.88±6.90**	2.68±0.11	0.91±0.45	**67.21±2.13**
500	ER	−10.42±3.42	1.02±5.55	−8.42±4.83	−3.12±4.02	0.56±2.52	65.52±1.56
	MER	−10.59±3.83	0.89±5.03	-	-	-	-
	GEM	−10.59±3.26	0.11±5.66	−12.53±0.65	1.36±3.05	0.17±0.59	54.19±2.37
	A-GEM	−9.74±3.60	1.10±7.30	−6.38±8.64	**6.36±3.88**	0.03±1.20	52.50±0.51
	iCaRL	N/A	N/A	N/A	N/A	-	-
	FDR	−9.27±2.80	4.73±5.08	**−6.23±8.79**	3.71±2.70	−0.32±0.43	65.97±1.02
	GSS	−10.16±3.48	0.17±5.32	−7.84±4.43	2.11±3.31	**0.89±0.94**	58.19±4.42
	HAL	−9.02±5.06	0.79±7.26	−7.15±7.57	3.06±1.03	1.33±0.23	64.21±3.16
	DER	**−7.96±2.57**	**1.17±6.37**	−13.26±1.08	−4.52±2.39	0.21±1.21	**72.45±0.14**
	DER++	−10.90±4.88	−2.92±5.32	−6.29±8.89	−0.31±1.86	−0.35±0.01	67.05±0.11
5120	ER	−10.97±3.70	0.17±3.46	−8.45±10.75	−1.05±5.87	**1.46±1.15**	**73.03±1.59**
	MER	−10.50±3.35	−0.33±5.81	-	-	-	-
	GEM	−9.51±3.83	−0.28±9.16	−9.18±4.27	−1.24±0.83	1.03±0.89	62.06±3.01
	A-GEM	−11.31±3.44	**1.14±7.08**	−8.01±6.31	−3.94±0.82	0.43±0.39	51.05±1.34
	iCaRL	N/A	N/A	N/A	N/A	-	-
	FDR	**−9.25±4.65**	−1.30±5.90	−7.69±5.95	−0.52±0.54	−0.13±0.54	72.54±0.35
	GSS	−10.89±3.52	−2.19±6.64	−9.88±2.21	−0.13±5.24	0.34±1.49	63.39±4.55
	HAL	−10.06±4.46	0.16±7.43	−10.34±3.22	0.32±1.09	0.52±0.47	66.00±0.09
	DER	−11.59±4.34	−2.42±5.22	**−5.98±8.44**	2.37±3.98	0.32±0.18	71.12±0.53
	DER++	−10.71±2.95	0.20±9.44	−11.23±2.67	**4.56±0.02**	0.06±0.22	72.11±1.81

Tab. 4.6: FFWD results for the Experiments of Sec. 4.3.

		FBWD					
$\|\mathcal{M}\|$	Method	S-MNIST		S-CIF10		P-MNIST	R-MNIST
		Class-IL	Task-IL	Class-IL	Task-IL	Domain-IL	Domain-IL
–	FT	−99.10±0.55	−4.98±2.58	−96.39±0.12	−46.24±2.12	−57.65±4.32	−20.34±2.50
–	oEWC	**−97.79±1.24**	−0.38±0.19	**−91.64±3.07**	−29.13±4.11	−36.69±2.34	−24.59±5.37
	SI	−98.89±0.86	−3.46±1.69	−95.78±0.64	−38.76±0.89	**−27.91±0.31**	**−22.91±0.26**
	LwF	−99.30±0.11	−6.21±3.67	−96.69±0.25	−32.56±0.56	-	-
	PNN	-	**0.00±0.00**	-	**0.00±0.00**	-	-
200	ER	−21.36±2.46	−0.82±0.41	−61.24±2.62	−7.08±0.64	−22.54±0.95	−8.24±1.56
	MER	−20.38±1.97	−0.81±0.20	-	-	-	-
	GEM	−22.32±2.04	−1.14±0.48	−82.61±1.60	−9.27±2.07	−29.38±2.56	−11.51±4.75
	A-GEM	−66.15±6.84	**−0.06±2.95**	−95.73±0.20	−16.39±0.86	−31.69±3.92	−19.32±1.17
	iCaRL	**−11.73±0.73**	−0.23±0.06	**−28.72±0.49**	**−1.01±4.15**	-	-
	FDR	−21.15±4.18	−0.50±0.19	−86.40±2.67	−7.36±0.03	−20.62±0.65	−13.31±2.60
	GSS	−74.10±3.03	−4.29±2.31	−75.25±4.07	−8.56±1.78	−47.85±1.82	−20.19±6.45
	HAL	−14.54±1.49	−0.48±0.20	−69.11±4.21	−11.91±0.52	−15.24±1.33	−11.71±0.26
	DER	−17.66±2.10	−0.56±0.18	−40.76±0.42	−6.21±0.71	−13.79±0.80	−5.99±0.46
	DER++	−16.27±1.73	−0.55±0.37	−32.59±2.32	−5.16±0.21	**−11.47±0.33**	**−5.27±0.26**
500	ER	−15.97±2.46	−0.36±0.20	−45.35±0.07	−3.54±0.35	−14.90±0.39	−7.52±1.44
	MER	−11.52±0.56	−0.44±0.17	-	-	-	-
	GEM	−15.47±2.03	−0.27±0.98	−74.31±4.62	−9.12±0.21	−18.76±0.91	−7.19±1.40
	A-GEM	−65.84±7.24	−0.54±0.20	−94.01±1.16	−14.26±4.18	−28.53±2.01	−19.36±3.18
	iCaRL	−11.84±0.73	**−0.25±0.09**	−25.71±1.10	**−1.06±4.21**	-	-
	FDR	−13.90±5.19	−1.27±2.43	−85.62±0.36	−4.80±0.30	−12.80±1.28	−6.70±1.93
	GSS	−60.35±6.03	−0.77±0.62	−62.88±2.67	−7.73±3.99	−23.68±1.35	−17.45±9.92
	HAL	−9.97±1.62	−0.30±0.26	−62.21±4.34	−5.41±1.10	−11.58±0.49	−6.78±0.87
	DER	−9.58±1.52	−0.39±0.18	−26.74±0.15	−4.56±0.45	−8.04±0.42	−3.41±2.18
	DER++	**−8.85±1.86**	−0.34±0.16	**−22.38±4.41**	−4.66±1.15	**−7.62±1.02**	**−3.18±0.14**
5120	ER	−6.07±1.84	0.03±0.36	−13.99±1.12	**0.08±0.06**	−5.24±0.13	−2.55±0.53
	MER	**−3.22±0.33**	0.05±0.11	-	-	-	-
	GEM	−4.14±1.43	0.16±0.85	−75.27±4.41	−6.91±2.33	−6.74±0.49	**−0.06±0.29**
	A-GEM	−55.04±10.93	**0.78±4.16**	−84.49±3.08	−9.89±0.40	−23.73±2.22	−17.70±1.28
	iCaRL	−11.64±0.72	−0.22±0.08	−24.94±0.14	−0.99±1.41	-	-
	FDR	−11.58±3.97	−0.87±1.66	−96.64±0.19	−1.89±0.51	−3.81±0.13	−2.81±0.47
	GSS	−7.90±1.21	−0.09±0.15	−58.11±9.12	−6.38±1.71	−19.82±1.31	−17.05±2.31
	HAL	−6.55±1.63	0.02±0.20	−27.19±7.53	−4.51±0.54	−4.27±0.22	−2.25±0.01
	DER	−4.53±0.83	−0.31±0.08	−10.12±0.80	−2.59±0.08	−3.49±0.02	−1.73±0.10
	DER++	−4.19±1.63	−0.13±0.09	**−6.89±0.50**	−1.16±0.22	**−2.93±0.15**	−1.18±0.53

Tab. 4.7: FBWD results for the Experiments of Sec. 4.3.

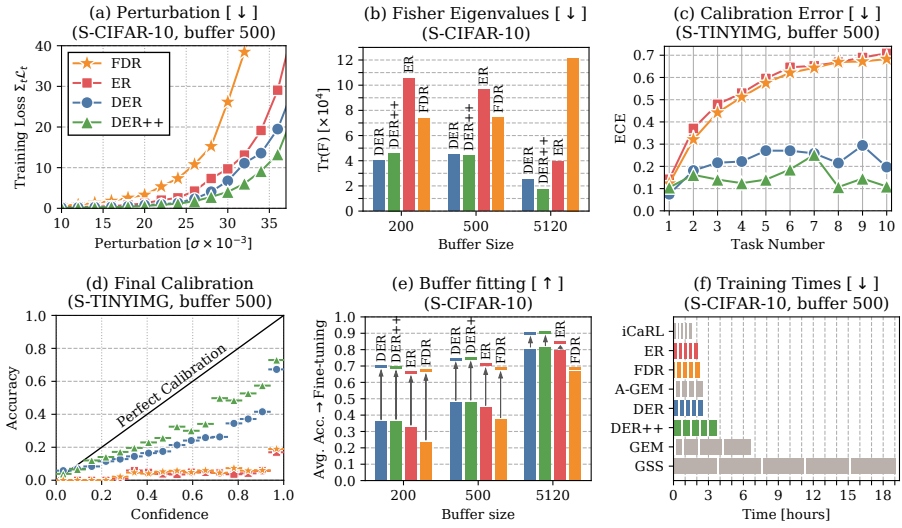


Fig. 4.1: Results for the model analysis. [↑] higher is better, [↓] lower is better.

We also observe that methods resorting to gradients (GEM, A-GEM, GSS) seem to be less effective in this setting, possibly due to gradients from different tasks being similar and hard to disentangle.

The gap in performance we observe in Domain-IL is also found in the Class-IL setting, as DER is remarkably capable of learning how classes from different tasks are related to each other. This is not so relevant in Task-IL, where DER performs on average on par with ER. In it, classes only need to be compared in exclusive subsets and maintaining an overall vision is not particularly rewarding. DER++ manages to effectively combine the strengths of both methods, resulting in generally better accuracy even in this scenario. Interestingly, iCaRL proves very effective when using a small buffer; we believe that this is due to its helpful *herding* strategy, ensuring that all classes are equally represented in memory.

4.4 Analysis

In this section, we provide an in-depth analysis of DER and DER++ by comparing them against FDR and ER. In doing so, we gather insights on the employment of logits sampled throughout the optimisation trajectory, as opposed to ones at task boundaries and ground-truth labels.

4.4.1 DER Converges to Flatter Minima

Recent studies [24, 65, 76] link DNN's capability for generalisation to the geometry of the loss function, namely the flatness of the attained minimum. While these works link flat minima to good train-test generalisation, here we are interested in examining their weight in CL.

Intuitively, if the optimisation converges to a sharp minimum w.r.t. $\mathcal{T}_{1...k}$, the attained solution will show little tolerance w.r.t. local perturbations. For this reason, we expect the drift produced in parameter space by fitting $\mathcal{T}_{k'}$ (for $k' > k$) to produce a serious drop in performance. The contrary happens if $\mathcal{T}_{1...k}$ is optimised by attaining a flat minimum of the loss function: the model has room for exploring the neighbouring regions of the parameter space – where one may find a new optimum for task k' – without experiencing a severe failure on tasks $1, \dots, k$.

We conjecture that the effectiveness of our proposal is linked to its ability to attain flatter and more robust minima, thus allowing for easier generalisation to yet unseen data. To validate this hypothesis, we characterise the flatness of the training minima of FDR, ER, DER and DER++ by means of two distinct metrics.

Firstly, as done in [205, 206], we consider the model at the end of training and add independent Gaussian noise with growing σ to each parameter. This allows us to evaluate its effect on the average loss across all training examples. As shown in Fig. 4.1a (S-CIF10, $|\mathcal{M}| = 500$), ER and FDR reveal higher sensitivity to perturbations than DER and DER++.

Furthermore, [24, 65, 76] propose measuring flatness by evaluating the eigenvalues of $\nabla^2_\theta \hat{\mathcal{L}}_{\text{CL}}$: sharper minima correspond to larger Hessian eigenvalues. At the end of training on S-CIF10, we compute the empirical Fisher Information Matrix $F = \sum \nabla_\theta \hat{\mathcal{L}}_{\text{CL}} \, \nabla_\theta \hat{\mathcal{L}}^T_{\text{CL}} / N$ w.r.t. the whole training set (as an approximation of the intractable Hessian [24, 81]). Fig. 4.1b reports the sum of its eigenvalues $\text{Tr}(F)$: as one can see, DER and especially DER++ produce the lowest eigenvalues, which translates into flatter minima thus confirming our intuition. It is worth noting that FDR's outlying large $\text{Tr}(F)$ in the case of $|\mathcal{M}| = 5120$ could be linked to its low performance in S-CIF10, Class-IL.

4.4.2 DER Produces More Calibrated Networks

Calibration is a desirable property for any statistical model which measures how much the confidence of its predictions corresponds to its accuracy. Ideally, we expect output distributions whose shapes mirror the probability of being correct, thus providing an immediate quantification of how much one can trust any prediction. Recent works highlight how modern DNNs – despite largely outperforming the models from a decade ago – are significantly less calibrated [52], as they tend to yield overconfident predictions [84]. In real-world applications, AI tools should support decisions

in a continuous and online fashion (*e.g.*, weather forecasting [18] or econometric analysis [48]); therefore, calibration represents an appealing property for any CL system aiming for employment outside of a laboratory environment.

Fig. 4.1c and Fig. 4.1d respectively show the value of the Expected Calibration Error (ECE) [118] during the training and the reliability diagram at the end of it for S-TinyImg. We observe that DER and DER++ achieve a lower ECE than ER and FDR without further application of *a-posteriori* calibration methods (*e.g.*, Temperature Scaling, Dirichlet Calibration, etc.). This means that models trained using Dark Experience are less overconfident and, therefore, easier to interpret. As a final remark, *Liu et al.* link this property to the capability to generalise to novel classes in a zero-shot scenario [101], which could translate into an advantageous starting point for the subsequent tasks for DER and DER++.

4.4.3 DER Constructs a More Informative Buffer

As network responses provide a richer description of the corresponding data-point than ground-truth labels, we posit that the effectiveness of DER can also result from the increased amount of knowledge contained in its memory buffer: when compared to the one built by ER, the former represents a more informative summary of the overall CL problem. To validate this intuition, we train a new learner only on data stored in the buffer and evaluate its resulting accuracy. We run this test using the memories produced by DER, ER, and FDR and show the test-set accuracy in Fig. 4.1e. We observe that DER and DER++ produce the highest performance, surpassing ER, and FDR. This is particularly evident for smaller buffer sizes, indicating that DER's buffer should be especially preferred in scenarios with severe memory constraints.

In addition to pure performance, we are also interested in quantifying the ease with which a model trained on the buffer can be specialised to an already seen task: this would be required if new examples from an old distribution could become available on the stream. To simulate this scenario, we sample 10 samples per class from the test set and then fine-tune the model on them with no regularisation; Fig. 4.1e reports the average accuracy on the remainder of the test set of each task: even in this benchmark, DER's buffer yields better performance than ER and FDR, providing additional insights on its representation capabilities.

4.4.4 Training time

When deploying a continual learner in the wild, one typically cares about reducing the overall processing time, since training needs to keep up with the rate at which data is made available on the stream. In this regard, we assess the performance of DER, DER++ and other RBMs in terms of

wall-clock time (seconds) at the end of the last task. To guarantee a fair comparison, we conduct all tests under the same conditions, running each benchmark on a Desktop Computer with an NVIDIA Titan X GPU and an Intel i7-6850K CPU. Fig. 4.1f reports the execution time measured on S-CIF10, further breaking down the time needed for each of 5 tasks. We conclude that DER has a comparable running time w.r.t. other RBMs such as ER, FDR, and A-GEM and is much faster than approaches such as GEM and GSS.

4.5 Conclusions

In this chapter, we introduced DER, a simple RBM which exploits self-knowledge distillation to retain past experience by storing previous model activations in the memory buffer. DER has a similar complexity to ER but attains superior results proving capable of outperforming the SOTA on several CL classification benchmarks, spanning multiple scenarios. By means of additional analysis, we highlighted some key properties possessed by our proposal, namely its tendency to attain flatter minima, achieve higher calibration and store richer information in its buffer.

The next chapter will focus on identifying some possible shortcomings of DER and subsequently proposing an improved version of our baseline model. The distillation-based approach to rehearsal pioneered in this chapter also constitutes the basis for our approach to pre-training preservation in CL, which will be the subject of Chap. 10.

Since its original publication, DER has gained large popularity as a competitor in CL literature, mostly thanks to its ease of implementation and reliability. Additionally, the original codebase for this work has grown into the *Mammoth* CL library[5], which is freely available and used for experiments by several published CL papers to date [106, 197, 17, 54, 94, 9, 46, 19].

[5] `https://github.com/aimagelab/mammoth`

Chapter 5

Past, Present and Future in Knowledge Distillation Replay

5.1 Motivation

This chapter identifies and overcomes some limitations of DER, the simple RBM baseline proposed in Chap. 4. Specifically, we extend our previous method by allowing it to update its replay memory to absorb novel information regarding past data and to actively prepare for learning yet unseen classes. This results in an updated RBM called **eXtended Dark Experience Replay (X-DER)**, which we analyse in detail by means of thorough Class-IL experiments and extensive ablation studies.

To facilitate the discussion of the issues affecting DER, we first introduce a detailed terminology to describe the responses of a CL classification model. Subsequently, we provide an analysis of DER's limitations, paving the way for the proposal of X-DER.

5.1.1 Terminology

To allow for a simpler exposition, we assume in the following that all presented tasks consist of an equal number of classes ($|\mathcal{Y}| \triangleq |\mathcal{Y}_0| = |\mathcal{Y}_1| = \cdots = |\mathcal{Y}_T|$), which is in line with the majority of Class-IL benchmarks presented in Sec. 2.3. We introduce the following categorisation which splits the output space of the model at task $\mathcal{T}_c$ (see also Fig. 5.1):

- **Past Logits** ($\ell_{\text{pa}[c]}$, with $\text{pa}[c] \triangleq \{0, 1, \dots, c \cdot |\mathcal{Y}_i| - 1\}$): logits modelling the probabilities of classes observed **up to** the current task c;
- **Present Logits** ($\ell_{\text{pr}[c]}$, with $\text{pr}[c] \triangleq \{c \cdot |\mathcal{Y}_i|, \dots, (c+1) \cdot |\mathcal{Y}| - 1\}$): logits referring to the classes appearing in the **current** task c;

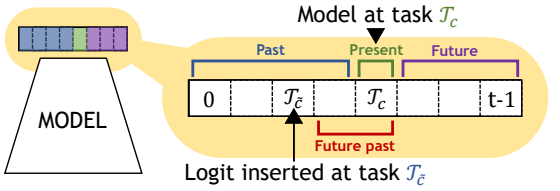


Fig. 5.1: An illustration depicting the Class-IL timeline. $\mathcal{T}_c$ is the task that is currently being learnt by the model; $\mathcal{T}_{\tilde{c}}$ indicates the task at which the example entered the memory buffer.

- **Future Logits** ($\ell_{\text{fu}[c]}$, with $\text{fu}[c] \triangleq \{(c+1)\cdot|\mathcal{Y}_i|,\dots,T\cdot|\mathcal{Y}|-1\}$): logits corresponding to **unseen** classes. They are not employed for classifying examples seen thus far but will become significant in the following tasks.

It is noted that the proportion of these partitions changes throughout training, as logits move from one partition to the other when changing task. **Only for buffer data-points**, we additionally identify **Future Past Logits**, an additional class of logits that comprises some **past** and **present logits**:

- **Future Past Logits** ($\ell_{\text{fp}[c;j]}$, with $\text{fp}[c;j] \triangleq \{j\cdot|\mathcal{Y}|,\dots,(j+1)\cdot|\mathcal{Y}|-1\}$, $j\in\{\tilde{c}+1,\dots,c\}$): given an exemplar $(x,y,\ell)\in\mathcal{M}$ stored at task $\tilde{c}$ ($\tilde{c}<c$), these logits model the classes of the j^{th} task discovered **after** the insertion of the example into the buffer.

In the remainder of this chapter, we use the expression **prediction head** to indicate a subset of contiguous logits within the classifier pertaining to the classes introduced in the same task. As we work in the Class-IL scenario, this should not suggest that distinct prediction heads within a classifier work independently, as the model's predicted probability always spans all seen classes.

We now present two key limitations affecting DER and DER++ w.r.t. their handling of future and future past logits in the memory buffer. A proposal for overcoming these issues is presented later in Sec. 5.2.

5.1.2 (L1): Future Past Blind Spot

Sec. 4.4.3 showed that DER and DER++ possess a richer and more informative memory buffer w.r.t. ER. This is due to their storing logits in its memory buffer, which not only encode the ground-truth label for each replay example, but rather a full probability distribution accounting for its similarity with other classes as well (the so-called **secondary information** [114]). However, upon close examination, we see that the information carried by these logits is limited to classes already seen at the time an example is inserted in $\mathcal{M}$.

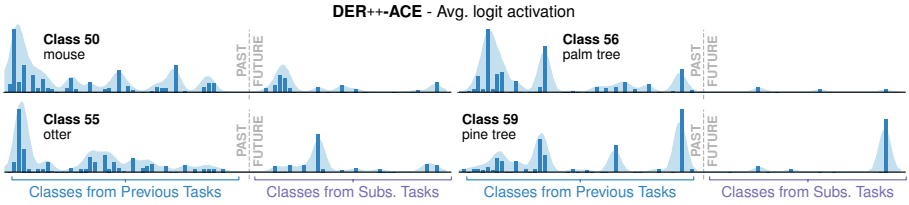


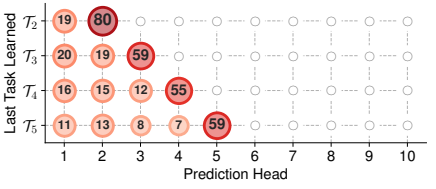
Fig. 5.2: An illustration of DER++'s blindness for future past classes. For the test examples belonging to four different classes introduced in $\mathcal{T}_4$ of S-CIF100, we show the average prediction at the end of the last task, omitting $\ell_{\text{pr}[4]}$. The model shows a clear prediction bias favouring classes shown in earlier tasks.

Indeed, when inserting an example in the rehearsal memory, we can reasonably assume that past and present logits encode all the information useful for later replay. However, by the time we move to subsequent tasks, the model discovers new classes and – with them – their relations with the ones learnt previously (*i.e.*, future past information). Unfortunately, DER and DER++ fail to capture this information and cannot learn these relations through replay, as the future past portion of their target distillation logits precedes the effective observation of the corresponding classes.

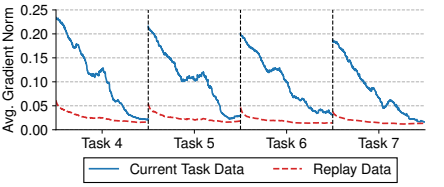
We illustrate this phenomenon experimentally in Fig. 5.2, which depicts for four classes of S-CIF100's $\mathcal{T}_4$ (*i.e.*, "*mouse*", "*palm tree*", "*otter*" and "*pine tree*") the average predictions produced by the model on the test set of S-CIF100 at the end of the last task, omitting the logits that refer to classes introduced at $\mathcal{T}_4$ to focus exclusively on secondary information. For each class, we observe that DER++-ACE[1] mostly emphasises the relations with classes that belong to previous tasks. It should be noted that such a result cannot be attributed to a particular choice of the order in which classes are encountered, as shown in the additional results that we present in Sec. 5.4.1 (Fig. 5.8 reports a reversed-class-order version of this experiment showing that DER++-ACE is *vice versa* led to emphasise the secondary information w.r.t. classes that are here neglected as they belong to subsequent tasks).

As a final note, we remark that this limitation does not apply to those distillation-based models that use previous network checkpoints to compute the regularisation objective [96, 138, 60]. In fact, as the teacher is updated at every task boundary, its responses change and come to include future past logits. The downside of this strategy, however, is that the update concerns past logits as well as future past ones, making the teaching signal vulnerable to forgetting (newer teachers struggle on old tasks).

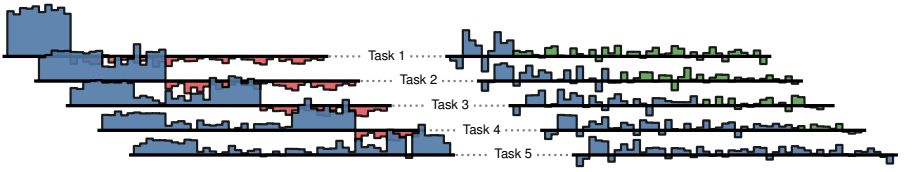
[1]In this experiment, we equip DER++ with Asymmetric Cross-Entropy (ACE) [19], to compensate the L2 bias issue described in the next section (otherwise, its effect would overshadow L1).



(a) Considering only examples of previous S-CIF100 tasks misclassified by DER++, the percentages of predictions won by each prediction head, revealing a clear bias towards the present. Later tasks are omitted as they entail the same issue.



(b) Average norm of DER++ gradients w.r.t. stream and buffer data on S-CIF100: the contribution of new examples on the stream significantly outweighs the one of replay items.



(c) Average logit values stored in the memory buffer by DER++ (left) and X-DER (right) for the first 50 classes of S-CIF100. For DER++, future logits (*red*) are strongly biased towards negative values. Thanks to the future preparation adopted by X-DER, its future logits (*green*) are comparable to past ones (*blue*).

Fig. 5.3: Three illustrations highlighting bias in DER++.

5.1.3 (L2): Overemphasis of Present Logits

The accumulation of bias towards present classes has a clear negative impact on performance, as investigated in Sec. 3.3.2 and also confirmed by several recent works [182, 3, 19, 114]. This issue also affects DER and DER++: similarly to [182], we can quantitatively characterise it by evaluating how predictions distribute across different prediction heads (as training progresses). In particular, we focus our analysis on misclassified examples belonging to tasks prior to the current one and highlight in Fig. 5.3a that the (wrong) prediction predominantly ends up in the last observed task.

The negative bias towards past classes can be ascribed to the optimisation of the cross-entropy loss on examples from the current task. As pointed out in [19], when a new task is presented to the net, an asymmetry arises between the contributions to weights updates of replay data and current examples: the gradients of new (poorly fit) examples prevail (Fig. 5.3b). If we only aim at learning the current task, this is desirable as it favourably dampens logits of past classes and prevents confusion. However, a hasty attenuation of earlier classes clashes with the goal of producing a unified classifier.

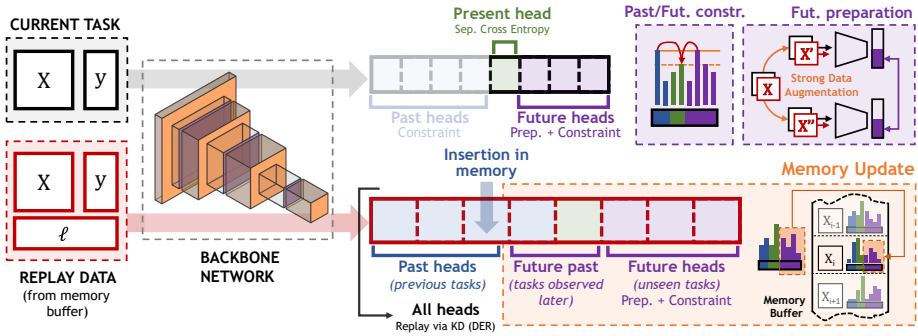


Fig. 5.4: X-DER uses distinct objectives for different partitions of the output space: *i)* it applies Cross-Entropy in isolation on the head of the current task; *ii)* it relieves forgetting by applying Knowledge Distillation on examples from $\mathcal{M}$; *iii)* it warms future logits, tied to unseen classes. Predictions in $\mathcal{M}$ are updated to deal with secondary information (*future past*) relating old examples with the classes emerging later on the stream.

Similarly, we observe a consistent negative bias towards future classes. We again ascribe this behaviour to the cross-entropy loss: since its application spans all prediction heads, the future ones are always given zero post-softmax targets and are thus strongly pushed towards pre-softmax negative values. Again, this desirably prevents future classes from being predicted; however, we mean future heads to accommodate the learning of future tasks. Therefore – if the negative bias accumulates so strongly on these heads – the recovery from that situation may slow down and complicate the learning of new tasks. In this regard, Fig. 5.3c illustrates the behaviour of future logits and compares the average responses of both DER++ (Fig. 5.3c, left) and the new approach we propose in Sec. 5.2 (Fig. 5.3c, right). We observe that the former consistently exhibits negative values for unseen classes, whereas the latter avoids bias accumulation on the account of the regularisation it imposes on future logits.

5.2 eXtended Dark Experience Replay

By addressing the above-discussed limitations of DER and DER++, we propose an enhanced model called **eXtended Dark Experience Replay (X-DER)**. A visual overview of the proposed approach is presented in Fig. 5.4.

5.2.1 Future Past Update

To deal DER and DER++'s failure to capture future past secondary information, we devise a simple procedure designed to keep the memory buffer

Alg. 5.1: Update of future past logits

Input: memory buffer $\mathcal{M}$, buffered example $(x, y, \ell^{\mathcal{M}}) \in \mathcal{M}$
new logits ℓ, attenuation rate γ (default 0.8)
$\ell^{\mathcal{M}}_{\text{gt}} \leftarrow \text{one-hot}(y) \cdot \ell^{\mathcal{M}}$
$\ell_{\text{fpmax}} \leftarrow \max_{j \in \text{fp}[c;c]} \ell_j$ ▷ get maximum future past logit
$h \leftarrow \min(\gamma \cdot \ell^{\mathcal{M}}_{\text{gt}} / \ell_{\text{fpmax}}, 1)$ ▷ compute the rescaling factor
for j **in** $\text{fp}[c;c]$ **do**
  $\ell^{\mathcal{M}}_j \leftarrow h \cdot \ell_j$
end for
$\mathcal{M} \leftarrow \text{update-record}(\mathcal{M}, (x, \ell^{\mathcal{M}}, y))$ ▷ save new logits into $\mathcal{M}$

up to date. Let us suppose the model is learning task $\mathcal{T}_c$ and an example $(x, y, \ell^{\mathcal{M}}) \in \mathcal{M}$ from a previous task $\mathcal{T}_{\tilde{c}}$ is sampled from the memory buffer for replay. Current model output $\ell \triangleq h_\theta(x)$[2] now contains the secondary information of task $\mathcal{T}_c$ for x, which is missing in $\ell^{\mathcal{M}}$. Therefore, we propose to **implant** the corresponding logits $\ell_{\text{fp}[c;c]}$ into the memory entry containing $\ell^{\mathcal{M}}$. Such an operation only involves the present prediction head and is applied both while learning $\mathcal{T}_{\tilde{c}}$ and at the end of it.

From a technical perspective, we do not simply overwrite previous logits with new ones. Since nothing prevents the value range of logits from changing in subsequent tasks, simply implanting their values in the memory buffer could produce unstable targets[3]. especially relevant if we consider and using these for later replay would exacerbate the issue even more. Instead, we carefully re-scale the portion tied to future past so that its maximum logit $\ell_{\text{fpmax}} = \max_{j \in \text{fp}[c;c]} \ell_j$ is lower than the ground-truth one $\ell^{\mathcal{M}_{\text{gt}}} = \text{one-hot}(y) \cdot \ell^{\mathcal{M}}$ already in memory. Formally:

$$\ell^{\mathcal{M}}_k \longleftarrow \ell_k \cdot \min(\gamma \frac{\ell^{\mathcal{M}}_{\text{gt}}}{\ell_{\text{fpmax}}}, 1), \quad k \in \text{fp}[c;c] \tag{5.1}$$

where $\gamma \in [0, 1]$ is a hyper-parameter controlling the attenuation rate. For a in-detail exposition of the update procedure, please refer to Alg. 5.1.

5.2.2 Future Preparation

Most Class-IL methods exploit the information available up to the current task to prevent the leak of past knowledge. Here, we take an extra step and argue that the same care should be placed on preparing prediction heads to accommodate future classes. The underlying intuition is illustrated in Fig. 5.5: considering JT on all tasks (Fig. 5.5, left) as the optimal solution we have to approximate, standard CL approaches (Fig. 5.5, centre) focus

[2] With h_θ indicating pre-softmax model responses of $f_\theta(\cdot)$, as defined in Sec. 3.3.2.

[3] For instance, the accumulation of bias highlighted in Sec. 5.1.3 is likely to produce targets whose future-past portion significantly overshoots the present.

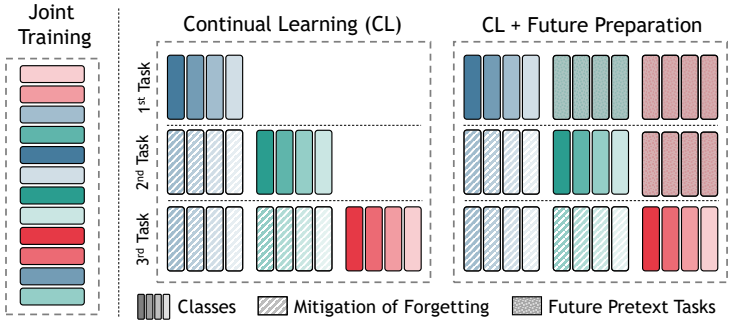


Fig. 5.5: X-DER introduces pretext tasks for anticipating unseen classes.

only on a (growing) part of the overall problem, *i.e.*, the tasks seen up to the current one. Instead, we claim that even a coarse guess on unseen tasks can lead to a better estimate of the overall CL objective (Eq. 2.1).

X-DER pursues this goal directly through optimisation (Fig. 5.5, right) by encouraging the (yet-to-be-established) semantics of logits corresponding unseen classes to be consistent across instances of the same class. As outlined by the field of contrastive self-supervised learning [31, 198], the skilful use of data augmentation techniques can produce useful representations even when no label is made available to the learner. As such information is not available for future classes while training, we expect a contrastive objective to be an effective warm-up strategy for future tasks.

Intuitively, the auxiliary objective we present in the following encourages each future prediction head to yield *similar responses* for *similar examples*. However, as we are given the labels of both the examples of the current task and the memory buffer, we refine the contrastive objective by incorporating class supervision. As shown in [77], we can leverage it to pull together representations of examples from the same class and to do the opposite for different classes.

In practice, given a batch of N examples, we extend it by appending N additional views of the original items (obtained through strong data augmentation). We then consider the model response $\ell(x_i) \triangleq h_\theta(x_i)$ for the i^{th} example: in particular, we firstly focus on the (normalised) j^{th} future prediction head (s.t. $j \in \{c{+}1, \dots, T{-}1\}$), which we denote with $\tilde{\ell}_{\text{fu}[c;j]}(x_i) \triangleq \text{L2Norm}(\ell_{\{j\cdot|\mathcal{Y}|,\dots,(j+1)\cdot|\mathcal{Y}|-1\}}(x_i))$. We then compute the following loss term:

$$\mathcal{L}_{\text{SC}}(x_i, y_i; j) = -\sum_{p \in P(i)} \log \frac{\exp(\tilde{\ell}_{\text{fu}[c;j]}(x_i) \cdot \tilde{\ell}_{\text{fu}[c;j]}(x_p)/\tau)}{\sum_{\substack{k=1 \\ k \neq i}}^{2N} \exp(\tilde{\ell}_{\text{fu}[c;j]}(x_i) \cdot \tilde{\ell}_{\text{fu}[c;j]}(x_k)/\tau)}, \tag{5.2}$$

where $P(i) = \{p \in \{1, \dots, 2N\} \; ; \; i \neq p \wedge y_i = y_p\}$ stands for the positive set (*i.e.*, the indices of examples sharing the label of the i^{th} item) and

τ is a positive scalar value that acts as a temperature. The full future preparation objective is simply obtained by further averaging the values of Eq. 5.2 across all future heads:

$$\mathcal{L}_{\text{FP}}(x_i, y_i) = \sum_{j=c+1}^{T-1} \mathcal{L}_{\text{SC}}(x_i, y_i; j). \tag{5.3}$$

Since Eq. 5.3 encourages unused heads to convey additional semantics about the examples seen so far, we find it beneficial to also include future logits in replay. Moreover, as new classes emerge in later tasks, we account for the corresponding semantic drift by applying the update procedure outlined in Sec. 5.2.1 on these logits also.

5.2.3 Bias Mitigation

Sec. 5.1.3 presents the accumulation of bias on the current prediction head as a key weakness of DER and DER++. As also observed in other recent works [3, 114, 19], this issue can be mitigated by revising the way the cross-entropy loss is applied during training. Given an example from the current task, we do not compute the softmax activation over all logits, but rather restrict its application to those modelling the scores of current-task classes. This removes the predictions of past classes from the equation and avoids their penalisation by the outweighing gradients of novel examples. In formal terms, we compute the following objective:

$$\mathcal{L}_{\text{S-CE}}(x_i, y_i) = \text{CE}(\text{softmax}(\ell_{\text{pr[c]}}(x_i)), y_i). \tag{5.4}$$

While this modification has an important effect on the input stream, it is not strictly necessary when working on buffered examples of previous tasks assuming that they are class-balanced; for this reason, we apply Eq. 5.4 on the input stream and instead apply the softmax across the logits of all seen classes for replay data:

$$\mathcal{L}_{\mathcal{M}\text{-CE}}(x_i, y_i) = \text{CE}(\text{softmax}(\ell_{\text{pa[c]}}(x_i)), y_i). \tag{5.5}$$

As a side effect of the introduction of $\mathcal{L}_{\text{S-CE}}$ and $\mathcal{L}_{\mathcal{M}\text{-CE}}$, we are not training all prediction heads at once: nothing prevents past and future responses from overshooting present ones and causing trivial classification errors. We address this issue by introducing an extra optimisation constraint limiting the activation of past and future heads: we require their values for current-task examples to be lower than a safeguard threshold, identified as the logit $\ell_{\text{gt}}(x_i) \triangleq \text{one-hot}(y_i) \cdot \ell(x_i)$ corresponding to the ground-truth class. In doing so, we penalise the maximum past (future) logit $\ell_{\text{pa-max}}(x_i) = \max_{j \in \text{pa[c]}} \ell_j(x_i)$ ($\ell_{\text{fu-max}}(x_i) = \max_{j \in \text{fu[c]}} \ell_j(x_i)$) if

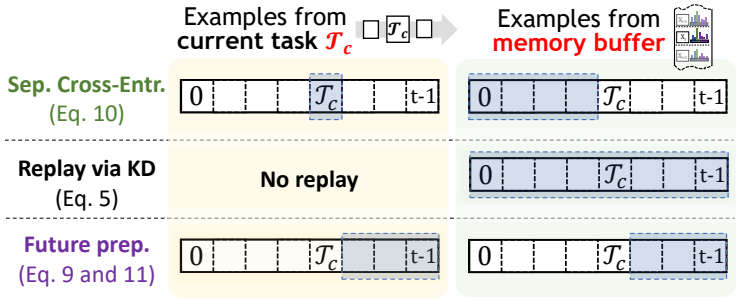


Fig. 5.6: Overview of which loss terms cover which prediction heads.

it oversteps $\ell_{\text{gt}}(x_i)$:

$$\begin{aligned}\mathcal{L}_{\text{PFC}}(x_i, y_i) = &\max(0, \ell_{\text{pa-max}}(x_i) - \ell_{\text{gt}}(x_i) + m) + \\ &\max(0, \ell_{\text{fu-max}}(x_i) - \ell_{\text{gt}}(x_i) + m),\end{aligned} \tag{5.6}$$

where m is a hyper-parameter that controls the strictness of the penalty.

5.2.4 Overall Objective

To sum up, **eXtended Dark Experience Replay (X-DER)** optimises the following loss as a proxy of Eq. 2.1:

$$\mathcal{L}_{\text{X-DER}} \triangleq \mathcal{L}_{\text{DER}} + \mathcal{L}_{\text{S}/\mathcal{M}\text{-CE}} + \mathcal{L}_{\text{F}}, \tag{5.7}$$

where $\mathcal{L}_{\text{DER}}$ is given in Eq. 4.4, $\mathcal{L}_{\text{S}/\mathcal{M}\text{-CE}}$ regroups the application of Eq. 5.4 and 5.5 on input and buffer data respectively:

$$\mathcal{L}_{\text{S}/\mathcal{M}\text{-CE}} = \mathop{\mathbb{E}}_{\substack{(x,y)\sim\mathcal{T}_c \\ (x',y')\sim\mathcal{M}}} \left[\mathcal{L}_{\text{S-CE}}(x, y) + \beta \cdot \mathcal{L}_{\mathcal{M}\text{-CE}}(x', y')\right], \tag{5.8}$$

and $\mathcal{L}_{\text{F}}$ regroups Eq. 5.3 and 5.6:

$$\mathcal{L}_{\text{F}} = \mathop{\mathbb{E}}_{\substack{(x,y)\sim\mathcal{T}_c \\ (x',y')\sim\mathcal{M}}} \Big[\lambda \underbrace{\mathcal{L}_{\text{FP}}(x||x', y||y')}_{\substack{\text{Eq. 5.3} \\ \text{Future Preparation}}} + \eta \underbrace{\mathcal{L}_{\text{PFC}}(x||x', y||y')}_{\substack{\text{Eq. 5.6} \\ \text{Past/Future Constraint}}}\Big]. \tag{5.9}$$

While outlining these objectives we introduce the additional hyper-parameters β, λ and η governing the contribution of each term to the overall loss. For the sake of clarity, Fig. 5.6 proposes a visual breakdown of the loss terms and the involved partitions of the classifier. For a deeper technical understanding of X-DER, we refer the reader to the pseudo-code provided in Alg. 5.2.

Alg. 5.2: eXtended Dark Experience Replay (X-DER)

Input: tasks $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_{T-1}$, parameters θ, learning rate lr,
buffer $\mathcal{M}$, scalars $\alpha, \beta, \lambda, \eta, m$, data aug/strong-aug routines.
$\mathcal{M} \leftarrow \{\}$
for c **in** $0, 1, \dots, T-1$ **do**
  for x, y **in** $\mathcal{T}_c$ **do**
    Sep. Cross Entropy (Eq. 10 and 13)
    $x^{\mathcal{M}}, y^{\mathcal{M}}, \ell^{\mathcal{M}} \leftarrow \text{sample}(\mathcal{M})$
    $\ell, \ell' \leftarrow h_\theta(\text{aug}(x)), h_\theta(\text{aug}(x^{\mathcal{M}}))$
    $\mathcal{L}_{\text{S-CE}} \leftarrow \text{CE}(\text{softmax}(\ell_{\text{pr[c]}}), y) + \beta \cdot \text{CE}(\text{softmax}(\ell'_{\text{pa[c]}}), y^{\mathcal{M}})$
    $\mathcal{M} \leftarrow \text{logits-update}(\mathcal{M}, (x^{\mathcal{M}}, y^{\mathcal{M}}, \ell^{\mathcal{M}}), \ell')$ ▷ see Alg. 5.1
    Dark Experience Replay (Eq. 5)
    $x^{\mathcal{M}}, y^{\mathcal{M}}, \ell^{\mathcal{M}} \leftarrow \text{sample}(\mathcal{M})$
    $\ell' \leftarrow h_\theta(\text{aug}(x^{\mathcal{M}}))$
    $\mathcal{L}_{\text{DER}} \leftarrow \alpha \left\| \ell^{\mathcal{M}} - \ell' \right\|_2^2$
    $\mathcal{M} \leftarrow \text{logits-update}(\mathcal{M}, (x^{\mathcal{M}}, y^{\mathcal{M}}, \ell^{\mathcal{M}}), \ell')$
    Future preparation (Eq. 9)
    $x^{\mathcal{M}}, y^{\mathcal{M}}, \_ \leftarrow \text{sample}(\mathcal{M})$
    $\mathcal{X}, \mathcal{Y} \leftarrow [x, x^{\mathcal{M}}], [y, y^{\mathcal{M}}]$
    $\mathcal{X}, \mathcal{Y} \leftarrow [\text{str-aug}(\mathcal{X}), \text{str-aug}(\mathcal{X})], [\mathcal{Y}, \mathcal{Y}]$
    $\mathcal{L}_{\text{FP}} \leftarrow \mathcal{L}_{\text{FP}}(\mathcal{X}, \mathcal{Y})$
    Past/Future Constraints (Eq. 11)
    $x^{\mathcal{M}}, y^{\mathcal{M}}, \_ \leftarrow \text{sample}(\mathcal{M})$
    $\mathcal{X}, \mathcal{Y} \leftarrow [x, x^{\mathcal{M}}], [y, y^{\mathcal{M}}]$
    $\ell \leftarrow h_\theta(\text{aug}(\mathcal{X}))$
    $\ell_{\text{gt}} \leftarrow \{\text{one-hot}(\mathcal{Y}[n]) \cdot \ell[n]\}_{n=1}^{|\mathcal{X}|}$ ▷ logits of ground-truth class
    $\ell_{\text{pa-max}} \leftarrow \{\max_{j \in \text{pa[c]}} \ell[n]_j\}_{n=1}^{|\mathcal{X}|}$ ▷ maximum past logits
    $\ell_{\text{fu-max}} \leftarrow \{\max_{j \in \text{fu[c]}} \ell[n]_j\}_{n=1}^{|\mathcal{X}|}$ ▷ maximum future logits
    $\mathcal{L}_{\text{PFC}} \leftarrow \max(0, \ell_{\text{pa-max}} - \ell_{\text{gt}} + m) + \max(0, \ell_{\text{fu-max}} - \ell_{\text{gt}} + m)$
    $\mathcal{L}_{\text{F}} \leftarrow \lambda \mathcal{L}_{\text{FP}} + \eta \mathcal{L}_{\text{PFC}}$
    $\mathcal{L}_{\text{X-DER}} \leftarrow \mathcal{L}_{\text{S-CE}} + \mathcal{L}_{\text{DER}} + \mathcal{L}_{\text{F}}$ ▷ overall loss (Eq. 12)
    $\theta \leftarrow \theta - lr \cdot \nabla_\theta \mathcal{L}_{\text{X-DER}}$ ▷ gradient step
  end for
  Insertion of the new items into $\mathcal{M}$
  $x, y \leftarrow \text{sample}(\mathcal{T}_c, \text{num-items} = |\mathcal{M}|/c + 1)$
  $\ell \leftarrow h_\theta(\text{aug}(x))$
  $\mathcal{M} \leftarrow \text{remove-items}(\mathcal{M}, \text{num\_items} = |\mathcal{M}|/c + 1)$
  $\mathcal{M} \leftarrow \mathcal{M} \cup (x, y, \ell)$
end for

5.3 Experiments

5.3.1 Experimental Settings

In this section, we present a comprehensive suite of Class-IL experiments aimed at evaluating the merits of X-DER against the SOTA. With respect to previous chapters, we choose to employ longer and more complex CL benchmarks such as S-CIF100, S-*mini*Img and the newly introduced action classification benchmark S-NTU60, aimed at evaluating the impact of catastrophic forgetting on non-image data (*i.e.*, skeletal graphs that expand in time) and Graph CNNs architectures [80]. All experimental settings are in accordance with Tab. 2.1 and results are presented in terms of FAA and FAF, averaged over 5 independent runs.

In addition to the usual upper and lower bound results provided by JT and FT respectively, we report the result for our baseline methods – DER and DER++ – and present the following Class-IL competitors: LwF.MC, ER, GDumb, ER-ACE, ER-RPC, BiC, iCaRL and LUCIR.

Since we also wish to validate the design choices of our proposal, we further compare against the four following **ablative variants** of X-DER:

- **X-DER without memory update (X-DER $^{\text{no mem}}_{\text{update}}$)**, which does not update logits through the sequence of tasks;
- **X-DER without future heads (X-DER $^{\text{w/out}}_{\text{future}}$)**, which handles new classes in the simplest manner possible – by adding a new prediction head only when the new task is presented;
- **X-DER with CE on future heads (X-DER $^{\text{CE}}_{\text{future}}$)**, a baseline that uses future heads like our proposal but – in line with what is done by DER and DER++ – includes them in the computation of the stream-specific portion of the separated cross-entropy loss thus targeting them with zero probabilities while training;
- **X-DER with RPC on future heads (X-DER $^{\text{RPC}}_{\text{future}}$)**, which does not require the semi-supervised learning objective of Sec. 5.2.2, but rather deals with future preparation by using the fixed **Regular Polytope Classifier** proposed in [132]. As detailed in Sec. 2.5, this approach ensures that prediction weights are equally distant by design.

5.3.2 Results

By examining Tab. 5.3.2, we can make the preliminary consideration that the considered regularisation approach (LwF.MC) consistently underperforms online RBMs[4]. This observation aligns with the results in previous

[4] *i.e.*, all RBMs but GDumb.

FAA (FAF)	S-CIF100		S-*mini*Img		S-NTU60
JT	70.44 (–)		53.55 (–)		85.75 (–)
FT	9.43 (89.82)		4.51 (77.38)		15.74 (92.85)
LwF.MC	16.22 (54.89)		12.20 (23.96)		28.24 (46.50)
$\lvert\mathcal{M}\rvert$	500	2000	2000	5000	500
ER	22.10 (73.64)	38.58 (53.28)	14.57 (64.49)	21.42 (50.36)	51.77 (48.54)
GDumb	9.98 (–)	20.66 (–)	15.22 (–)	27.79 (–)	27.59 (–)
ER-ACE	38.75 (40.04)	49.72 (25.71)	22.60 (23.74)	27.92 (19.72)	52.14 (23.33)
ER-RPC	22.34 (71.94)	38.33 (52.33)	15.60 (61.00)	24.69 (46.34)	49.40 (48.10)
BiC	36.02 (51.85)	46.39 (40.49)	12.96 (57.19)	14.45 (56.55)	29.20 (66.16)
iCaRL	46.52 (22.06)	49.82 (18.07)	22.58 (16.46)	22.78 (16.37)	45.83 (21.48)
LUCIR	40.59 (34.55)	41.73 (25.41)	14.97 (43.83)	17.61 (39.01)	58.06 (32.58)
DER	36.60 (54.99)	51.89 (34.54)	22.96 (48.78)	29.83 (36.38)	49.49 (43.09)
DER++	38.25 (50.54)	53.63 (33.66)	23.44 (46.69)	30.43 (37.11)	55.32 (35.95)
X-DER $^{\text{no mem update}}$	42.67 (24.03)	56.55 (9.24)	25.76 (16.76)	31.40 (13.50)	57.66 (12.52)
X-DER $^{\text{w/out future}}$	45.61 (33.31)	55.00 (22.94)	21.71 (36.92)	27.45 (18.39)	61.02 (9.80)
X-DER $^{\text{CE future}}$	47.67 (25.12)	55.61 (10.52)	27.18 (36.12)	30.69 (16.80)	61.58 (10.94)
X-DER $^{\text{RPC future}}$	48.53 (26.94)	57.00 (12.65)	26.38 (38.33)	29.91 (28.29)	62.41 (8.88)
X-DER	**49.93** (**19.90**)	**59.14** (**12.58**)	**28.19** (**20.45**)	**31.70** (**15.87**)	**64.86** (**9.95**)

Tab. 5.1: Experimental results on multiple Class-IL settings. Results reported as FAA and FAF (in parentheses).

chapters and in [6, 45], suggesting that the adoption of a replay memory is essential for achieving solid performance in Class-IL.

As it only learns from its memory buffer, the offline training of GDumb allows it to observe (few) examples from all classes jointly, avoiding issues related to bias by design. On S-*mini*Img, which features a long sequence of tasks, this is sufficient to outperform methods that do not compensate bias (*e.g.*, ER, DER, ER-RPC). However, since it entirely discards the remaining data from the input stream, GDumb produces a lower FAA w.r.t. to most online-learning methods.

Among ER-based approaches, ER-ACE stands out as the most effective thanks to its loss, carefully designed to prevent interference between the learning of the current task and the replay of old data. This trait allows to protect previously acquired knowledge, resulting in lower FAF.

On average, methods combining rehearsal and distillation achieve better performance w.r.t. simple replay. iCaRL limits forgetting consistently and achieves balanced accuracy on all seen tasks thanks to its nearest-mean-of-exemplars classifier. This is rewarding on the medium-length S-CIF100 benchmark but proves sub-optimal on both S-*mini*Img and S-NTU60 (due to forgetting on the former and to lack of fitting of the current task on the latter). Differently, LUCIR delivers a high accuracy

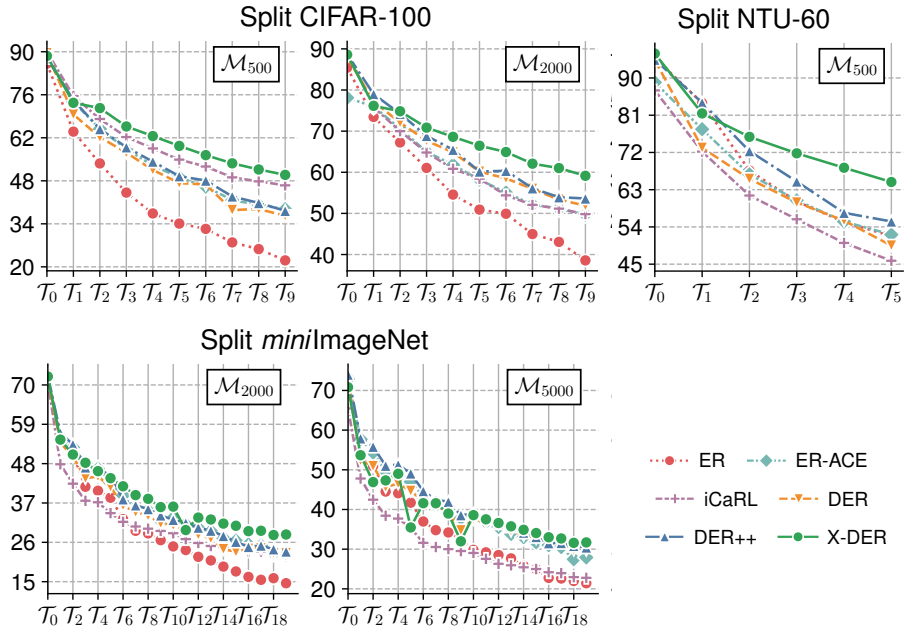


Fig. 5.7: For the experimental settings reported in Tab. 5.3.2, the average test-set accuracy after each observed tasks.

on the last few encountered tasks, proving very effective on the short S-NTU60 but struggling on longer sequences. While its performance is adequate on S-CIF100, BiC is characterised by the highest FAF on all other benchmarks, leading to a FAA score close to the one of LwF.MC.

Our previous proposals DER and DER++ qualify as strong baselines when combined with a large-enough memory buffer. However, due to the limitations explored in Sec. 5.1.2 and 5.1.3, they occasionally fall short of approaches that contrast bias more effectively (ER-ACE, iCaRL, LUCIR for $|\mathcal{M}| = 500$ on S-CIF100; ER-ACE and LUCIR on S-NTU60).

Compared to the current SOTA, X-DER delivers higher accuracy and lower forgetting across all benchmarks. As one can observe from a close exam of its incremental accuracy values (Fig. 5.7), the proposed enhancements lead to increased performance retention on past tasks, lifting its score over competitors significantly as training progresses.

To achieve deeper understanding, we further compare X-DER against its ablative baselines. By omitting to update the content of the memory buffer, X-DER $^{\text{no mem}}_{\text{update}}$ shows a significant drop in performance (especially relevant for smaller $|\mathcal{M}|$).

Comparatively, the strategy adopted for preparing future logits seems less influential. The proposed contrastive preparation loss of X-DER

yields the lowest FAF rates, validating our intuition to use past data to prepare future learning. Adopting the theoretically grounded but fixed design of X-DER $^{\text{RPC}}_{\text{future}}$ comes at a steady but non-negligible cost in performance across all benchmarks. X-DER $^{\text{CE}}_{\text{future}}$ shows that dampening future heads by indiscriminately applying CE leads to a further decrease in accuracy; however, even this approach is still preferable to X-DER $^{\text{w/out}}_{\text{future}}$, which produces higher FAF metrics. Its result stresses the importance of preparing the model for future classes and confirms that using future heads and replaying their logits acts as a remedy against forgetting.

5.4 Analysis

This section features an extended set of analytical experiments characterising X-DER w.r.t. the aspects presented in Sec. 4.4 and beyond.

5.4.1 X-DER Compensates Future Past Blindness

In this section, we assess whether the proposed X-DER is effective in dealing with the delicate issue discussed in Sec. 5.1.2. We repeat the preliminary experiment of Fig. 5.2 to record the the average output distribution delivered by DER++-ACE, X-DER and JT and report the detailed results in Fig. 5.8. For each model, we present two settings: in the first, classes are shown to the model in the usual order (*forward order*, blue bars); in the second, we reverse that order (*backward order*, orange bars).

Let us first focus on the class "mouse" (first panel) and consider the output of JT; this method does not learn continually but rather acts as an upper bound providing us with reference inter-class similarities. In the case at hand, it reveals that *mouse* examples mostly activate the logits associated with *hamster*s and *shrew*s. As expected, inverting the order of the classes does not alter the distribution produced by JT.

Instead, DER++-ACE shows differentiated behaviours: in forward order, it captures only the *mouse-hamster* similarity; *vice versa*, it learns about the *mouse-shrew* similarity only when trained in reverse order. This is a clear exemplification of limitation L1 at play: when training on classes in a forward order, *hamster*, *mouse* and *shrew* are learnt during the 3$^{\text{rd}}$, 5$^{\text{th}}$ and 7$^{\text{th}}$ incremental task of S-CIFl00 respectively. Accordingly, DER++-ACE replay targets include the association of *mouse* examples with high logits for the *hamster* class, but do not encode meaningful information regarding the *shrew* class, which is learnt in a later task. The opposite happens when the reverse order is applied (*shrew*s are learnt before *mice*).

Instead, X-DER shows a more coherent activation pattern when training in either order and is effective in capturing similarities with future classes as well as past ones. We primarily ascribe this behaviour to the update of future past targets in replay (Sec. 5.2.1).

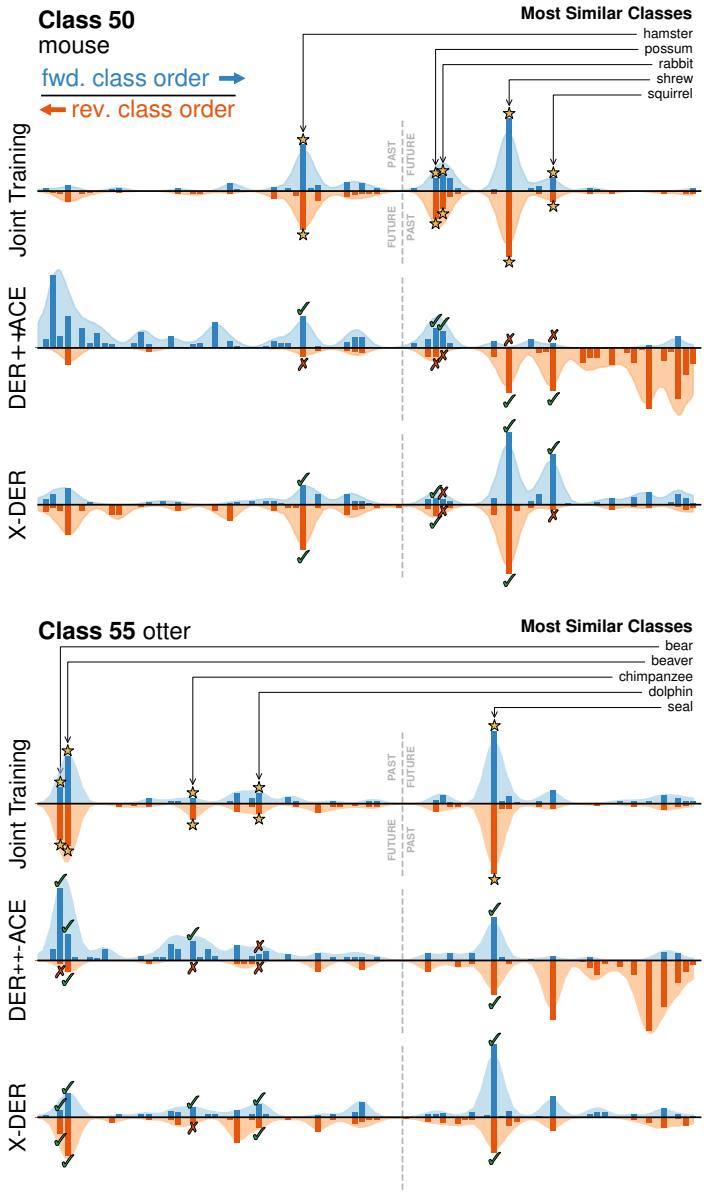


Fig. 5.8: Handling of future past logits – additional results.

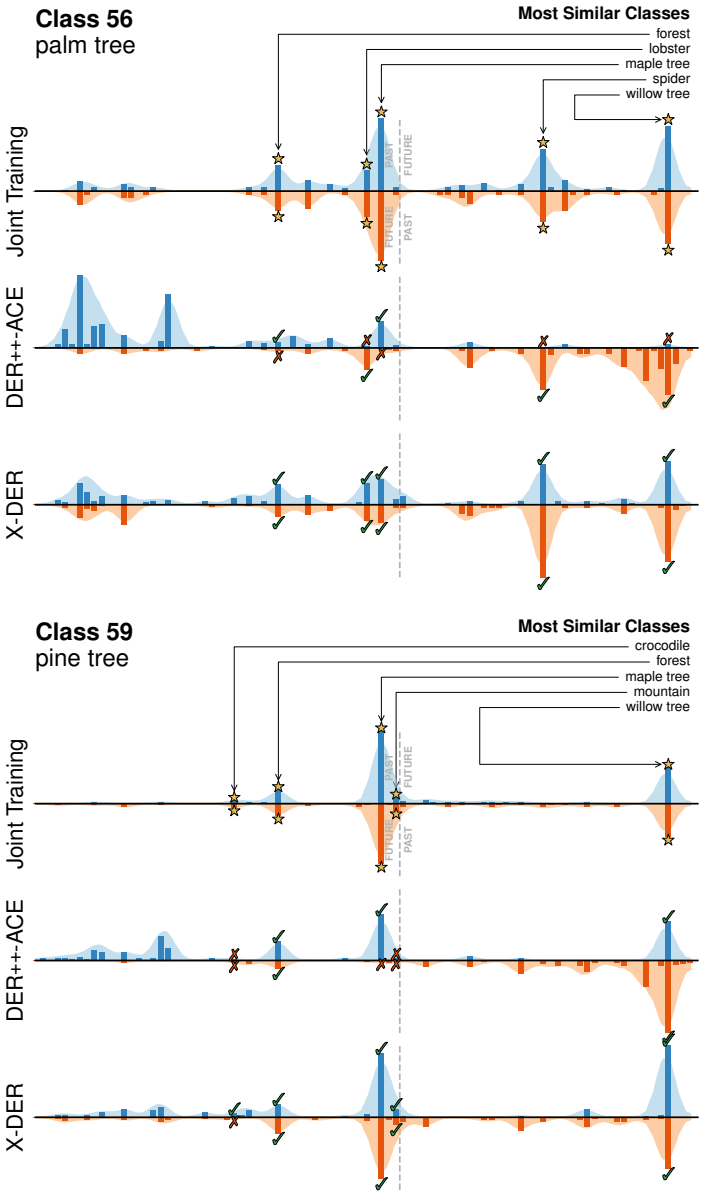


Fig. 5.8: Handling of future past logits – additional results. (cont.)

	KD	SS-ERR		SS-NLL	
$\|\mathcal{M}\|$		500	2000	500	2000
ER	–	0.71	0.68	4.00	4.32
ER-ACE	–	0.63	0.60	2.64	2.81
DER	✓*	0.67	0.64	2.22	2.21
DER++	✓*	0.67	0.64	2.22	2.25
LUCIR	✓	<u>0.60</u>	<u>0.59</u>	2.21	2.22
iCaRL	✓	<u>0.60</u>	0.60	<u>1.90</u>	<u>1.94</u>
X-DER $^{\text{no mem update}}$	✓*	0.64	0.61	2.14	2.10
X-DER	✓	**0.57**	**0.56**	**1.83**	**1.82**

Tab. 5.2: Secondary information metrics (lower is better). – indicates no use of Knowledge Distillation (KD) while training, ✓* indicates KD of past logits only, ✓ indicates KD of all logits (including future past).

5.4.2 X-DER Produces Better Continual Teachers

This section investigates the effectiveness of X-DER against forgetting of old tasks. We build upon the seminal work of [111], which has recently proposed a statistical approach to explaining Knowledge Distillation [59]. Essentially, the authors assume that the teacher's response $\mathbb{P}^t(y|x)$ constitutes an approximation of the true *Bayes class-probability distribution* $\mathbb{P}^*(y|x)$, which represents the suitability of each class y for a given input x (*i.e.*, an encoding of confusion among classes in model prediction). With respect to one-hot targets, it is proven that minimising the risk associated with $\mathbb{P}^*$ gives the student an objective with lower *variance*, which aids generalisation. However, the true $\mathbb{P}^*$ cannot be accessed and an imperfect estimate must be used instead (*e.g.*, the response of a teacher net). In that sense, *the better* the estimation of the true Bayes probabilities, *the higher* the generalisation capabilities of a student learning through the corresponding risk.

In the following, we leverage this newly introduced theory to shed light on key properties that characterise X-DER, namely the improved retention of secondary information, the increase informativeness of the constructed memory buffer and its calibration.

Analysis of Secondary Information

A compelling line of works analyses Bayes class-probabilities in the key **secondary information** [188, 114], *i.e.*, for each non-maximum prediction score, the model's belief about the semantic cues of the corresponding class within the input image. Unsurprisingly, *Yang et al.* identify the

preservation of secondary information as a key property of KD [188]: they empirically find that teachers with richer secondary information lead to students that generalise better. However – when dealing with catastrophic forgetting – it is problematic to capture rich secondary information, given that it only becomes available progressively as tasks advance.

Seeking to measure how effectively distinct CL approaches capture secondary information, we follow the setup proposed in [114] and – after training on S-CIF100 – we evaluate their predictive capability when ignoring the ground-truth class and instead using a coarse labelling, given by regrouping the 100 classes into their natural 20 super-classes. According to the authors of [114], a model achieving a high classification score in this setup proves more effective in retaining better secondary information, as classes belonging to the same super-class can be assumed to have higher visual similarity than the ones of different super-classes.

The retained secondary information can be quantified by two metrics [114]: on the one hand, the **Secondary-Superclass Error (SS-ERR)** equals 1 minus the probability of predicting the right super-class when the maximum logit is discarded during softmax computation; on the other, the **Secondary-Superclass NLL (SS-NLL)** considers the negative log-likelihood when using super-classes as labels.

The results in Tab. 5.2 show that X-DER, iCaRL and LUCIR consistently end up predicting the correct coarse classes (lower SS-ERR) and do so more confidently (lower SS-NLL). This is in line with our expectations: as these methods handle hindsight-learnt similarities between newly discovered classes and old ones, the corresponding teaching signal leads the student towards richer secondary information. In contrast, DER and DER++ yield lower metrics due to *i)* their distillation targets neglecting logits of future past, *ii)* the existence of a large bias towards the last seen classes. To verify the importance of *i)*, we also run this evaluation on the variant of X-DER that does not update its buffer logits (X-DER $^{\text{no mem}}_{\text{update}}$, Sec. 5.3.1); this expectedly results in metrics comparable to DER and DER++. ER – which applies no distillation at all – is also affected by issues *i)* and *ii)*; indeed, it produces the highest metrics among the evaluated methods. On the contrary, ER-ACE – which addresses *ii)* through its segregated objective – attains lower metrics closing in on DER and DER++. This highlights that bias control too plays a primary role in the capture and conservation of secondary information.

Offline Training on Memory Buffer

Distinct RBMs compared in Sec. 5.3.2 retain different summaries of the previously encountered knowledge: approaches such as ER, iCaRL and LUCIR keep labels of the recorded samples, DER and DER++ use the responses provided by the model at insertion time, while X-DER exploits responses that are updated as future past logits become available. As done

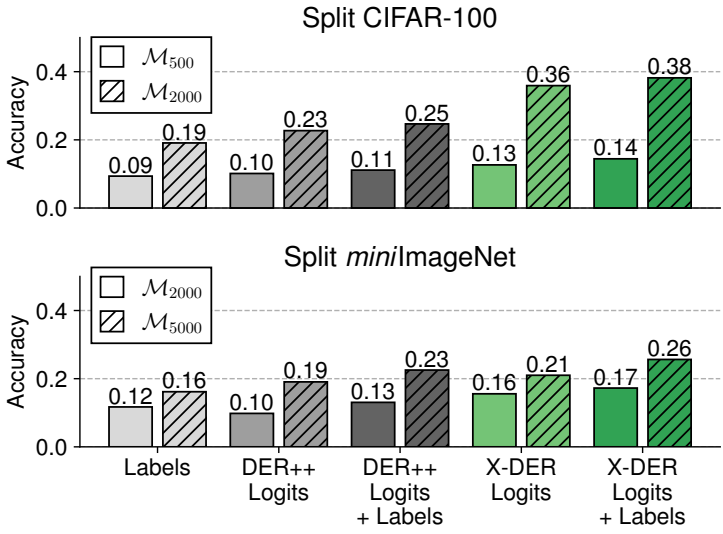


Fig. 5.9: Accuracy of models trained from scratch on memory buffers of ER (*Labels*), DER++ (*Logits*, *Both*), and X-DER (*Logits*, *Both*). The resulting accuracy measures the informativeness of the memory buffer.

previously in Sec. 4.4.3, we assess the amount of reliable information retained by these approaches by training a model from scratch using only the data available in the final buffers constructed by ER, DER++ and X-DER. We compute the performance achieved by the resulting models after training for 70 epochs and show the results on S-CIFl00 and S-*mini*Img in Fig. 5.9.

In line with the theoretical results of [111], we observe that relying on logits yields lower generalisation error w.r.t. learning from labels alone and combination of hard and soft supervision signals leads to slight improvements both for DER++ and X-DER. Most significantly, the use of updated logits of X-DER results in a steady improvement: when compared to DER++, we observe an average gain of 6% (when using logits alone) and 6.25% (combined with labels). Based on the considerations above, we attribute this additional regularisation effect to the exploitation of future past logits, which arguably drives the model towards a better estimate of the true Bayes class-probabilities.

Calibration of Continual Learners: A New Perspective

A key issue that needs solving is coming up with a way to assess the quality of the approximation of the $\mathbb{P}^*$. *Menon et al.* [111] suggest that a rough quantification can be obtained through ECE [52]. Remarkably, this

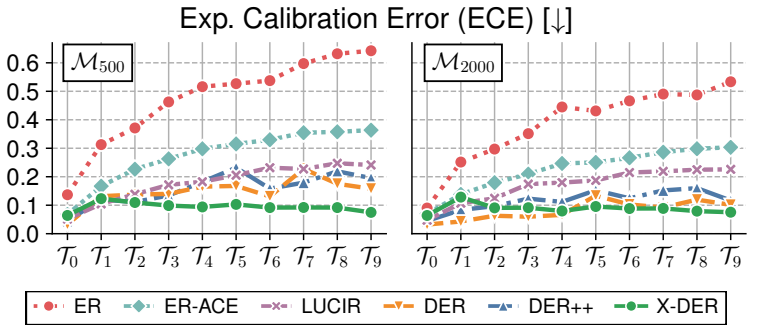


Fig. 5.10: Effect of several regularisation methods on net calibration (S-CIF100). While most of them degrade with lower memory size (left), X-DER yields robust performance.

provides a more meaningful interpretation to the experiment conducted in Sec. 4.4.2, in which we identified the higher degree of calibration of DER and DER++ as a key factor underlying their improved accuracy.

On these premises, we here repeat the evaluation on top of our new proposal. Fig. 5.10 shows the results obtained on S-CIF100: X-DER attains a lower ECE compared to other approaches. This not only applies to models replaying ground-truth labels such as ER, but also to DER and DER++, which use soft labels. This last finding further confirms our intuition that the improvements of X-DER can be linked to a better estimation of the underlying Bayes class-probabilities.

5.4.3 Future Preparation Matters

One of the key constituents of X-DER is the employment of a pretext task to warm up unused heads, leading to the *gentler* adaptation of the network to unseen data distributions and ultimately lowering the risk of forgetting. To verify whether this happens, we consider a setting where the model is given a few data-points of the incoming tasks to be used for few-shot adaptation and verify how well the feature space spanned by future heads trained as specified in Sec. 5.2.2 fulfils this purpose.

We consider several RBMs (including ER, DER, DER++ and our X-DER) and, firstly, stop their training after the 6^{th} task of S-CIF100. In Fig. 5.11a, we measure their performance on each of the remaining four tasks separately by fitting a Nearest Neighbour (NN) classifier on top of the activations given by the corresponding future prediction head, without finetuning the model on the new target data. We repeat this evaluation at varying training set sizes (ranging from one to fifty shots per class) and observe that X-DER achieves the best results among the tested approaches.

To paint a more comprehensive picture, we take into consideration

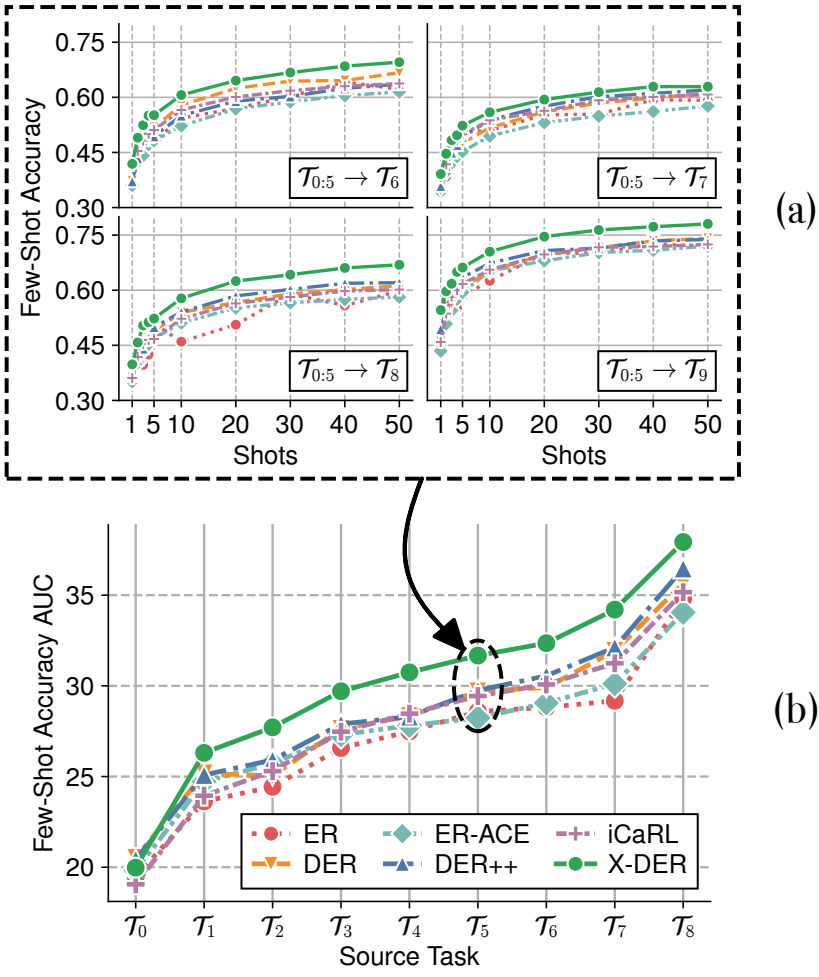


Fig. 5.11: Analysis of generalisation to unseen classes. (a) For each of the four remaining tasks of S-CIF100, the performance *vs* training-set size trend for different CL methods. (b) The curves describing the forward transfer at the end of every task.

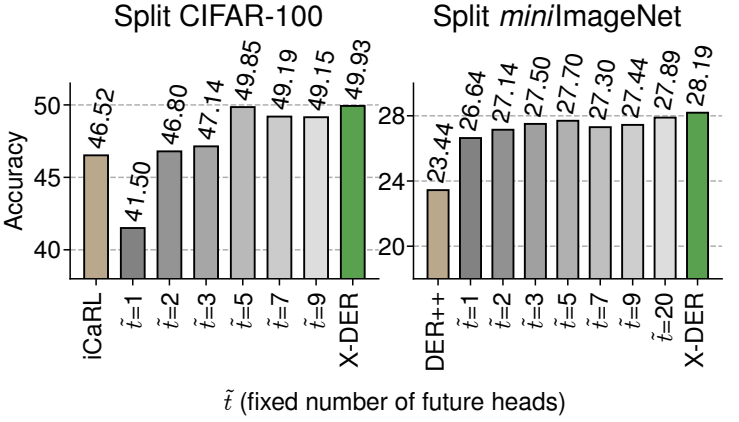


Fig. 5.12: On S-CIFl00 and S-*mini*Img, an analysis of how the number of heads pre-allocated in preparation of future tasks affects the FAA.

the transfer from all encountered task (and not only the 6^{th}) and assess how the capabilities linked to forward transfer evolve one task after the other. To do so, for each observed task $\mathcal{T}_t$ ranging from the first to the penultimate, we initially define the performance curves $\text{NN}_{t\to\tilde{t}}(k)$ over the unseen tasks $\tilde{t} > t$, where k indicates the number of shots per class (for reference, Fig. 5.11a depicts $\text{NN}_{5\to\tilde{t}}(k)$ $\forall \tilde{t} \in \{6,7,8,9\}$). Subsequently, we summarise each curve with the **Area Under the Curve** $\text{AUC}_{t\to\tilde{t}}$ and finally average the latter across $\tilde{t}$ (*e.g.*, $\text{AUC}_5 \triangleq \sum_{\tilde{t}=6}^{9} \text{AUC}_{5\to\tilde{t}}$), thus providing a compact generalisation measure w.r.t. all future tasks.

Fig. 5.11b then reports the trend of the AUC_t for the tested model. We do not observe a clear distinction in their performance on earlier tasks; however, the AUC curve of X-DER widens the gap as the number of seen tasks increases (it scales better to the number of seen tasks). This suggests that the more and more diverse the data present in the memory buffer, the higher the chances that optimising Eq. 5.3 will lead to good forward transfer on unseen data.

Pre-allocation of Future Tasks

In proposing X-DER, we have supposed that the overall number of tasks T can is known in advance. This allows us to instantiate a last fully-connected layer large enough to accommodate the logits for all seen and unseen classes. However, in practical scenarios, we may not know how many tasks will be encountered from the outset, bringing into question whether our approach can still be applied to those settings.

We here present a straightforward modification that enables the number

of future tasks to be unknown. We initially set up the last layer to expose $\tilde{t}+1$ prediction heads: precisely, the one dedicated to the first task and the remaining $\tilde{t}$ to future tasks. In addition, we instantiate a new head at the end of each task, so as to always have $\tilde{t}$ spare heads for incoming tasks.

Fig. 5.12 depicts how such a modification affects performance for varying $\tilde{t}$. We draw the following conclusions: *i)* given the slight gap in performance between X-DER and the proposed variant, knowing the overall number of tasks does not appear necessary for achieving good results; *ii)* a higher number of pre-allocated heads positively influences FAA. This latter finding suggests that future logits also play a role against forgetting: we conjecture that the rehearsal of future logits might represent an additional guard against forgetting even if they do not encode a prediction probability, as they still provide a reminder of past neural activities.

5.4.4 Why Local Minima Geometry Matters in CL

Effectiveness of Flat Minima in Continual Learning

In Sec 4.4.1, we reported some agreed-upon hypotheses on the relation between the nature of the local minima attained by a CL model and the thereby linked generalisation capabilities. Flatness around a loss minimiser is regarded as a remarkable property for CL settings: intuitively, a loss region tolerant towards local displacements favours later optimisation trajectories that entail a less severe drop in performance for old tasks.

As a proof of concept, we used two common metrics to characterise the geometry of the minima and verify that DER and DER++ exhibit favourably flatter minima. A similar approach was recently taken in [113], which assessed the impact of different training regimes on forgetting and showed that forgetting diminishes when strategies known to affect the width of the minima (*e.g.*, higher initial learning rates, dropout, small batch sizes, etc.) are applied. Since such a matter is still largely explained through intuition, we propose an empirical experiment showing the general importance of flat minima in CL. Given a sequence of two tasks, we deliberately drive the optimisation of the former towards a wider minimum. Differently from [113], we explicitly pursue this objective by introducing a tailored term in the loss function. In this regard, we evaluate two distinct approaches:

- **LFR** [186], which seeks to minimise the ℓ_1-norm of the loss gradients w.r.t. a *malign* example forged so that: *i)* it lies in the ε-neighbourhood centred on a given (benign) example; *ii)* it maximises the norm of the gradients. The authors prove that the robustness towards this kind of attack favourably relates to the flatness of the loss surface;

- **LLR** [136], which promotes loss smoothness around the local neigh-

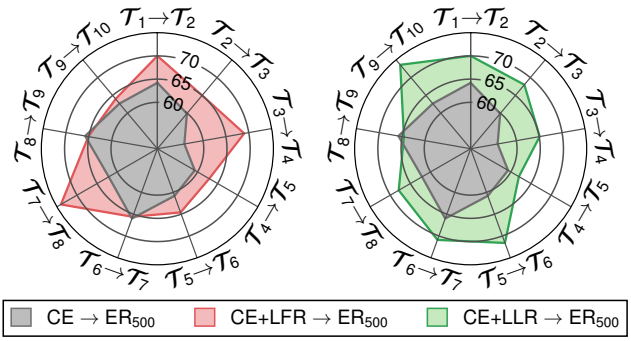


Fig. 5.13: For nine sequences of two tasks of S-CIF100, the final accuracy on the test-set of the former task. LFR and LLR, which encourage flatter minima, lead to higher retention of performance.

bourhood. As before, it consists of a regularisation term that depends on adversarial examples: supposing a smooth and approximately linear loss surface, the first-order Taylor expansion on w.r.t. these inputs should represent a good approximation of the value of the loss function. Consequently, LLR simply seeks to minimise the error one commits when using such an approximation.

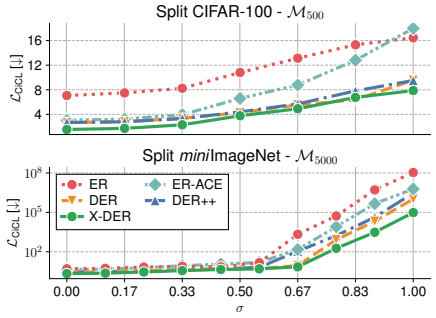
We train the network on the former task pairing the CE loss with either flattening regularisation and then measure the forgetting incurred by ER after the second task. As a baseline, we consider the results achieved when the additional regularisation is not applied during the former task. We conduct this evaluation on top of nine possible combinations of adjacent tasks of S-CIF100 and report the results in Fig. 5.13. The application of either regularisation technique upper-bounds the baseline across all tasks configurations.

This provides strong empirical evidence towards the benefits of attaining flat loss minimisers in CL and it corroborates the intuition behind the effectiveness of those self-distillation-based approaches (*e.g.*, iCaRL, LwF, DER and DER++, etc.) which are known to lead to such a regime [205, 204].

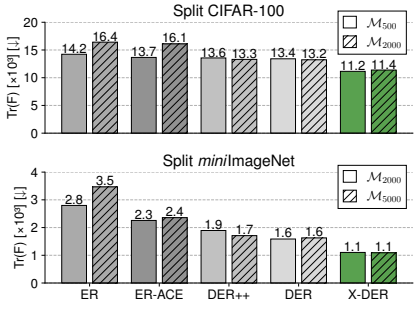
Measuring the Flatness

Having established the significance of flat minimisers in CL problems, we present here the two quantitative evaluations first introduced in Sec. 4.4.2 to illustrate the stability and flatness of the optima observed for X-DER and other CL approaches.

Firstly, we measure how weight perturbations affect the **expected loss**



(a) For increasing σ, the value assumed by the loss function when Gaussian noise is applied to network weights.



(b) The of the eigenvalues of the Fisher Information Matrix, which models the curvature of the loss function around the solution.

Fig. 5.14: Analysis of minima attained by distinct approaches.

over the entire CL problem $\hat{\mathcal{L}}_{\text{CL}}$ (Eq. 2.2) w.r.t. to the training set [121, 76]:

$$\mathcal{L}_\sigma \triangleq \sum_{i=0}^{T-1} \mathop{\mathbb{E}}_{\substack{(x,y)\sim\mathcal{T}_i \\ \tilde{\theta}\sim\mathcal{N}(\theta,\sigma)}} \Big[\hat{\mathcal{L}}_{\text{CL}}\big(f_{\tilde{\theta}}(x), y\big)\Big]; \tag{5.10}$$

specifically, we follow the hints of [93, 121] and weigh the perturbation according to the magnitude of parameters ($\sigma_i = \alpha|\theta_i|$), thus preventing degenerate solutions [121]. With reference to Fig. 5.14a, it can be seen that logit-replay based models such as DER, DER++ and X-DER consistently preserve a lower value for Eq. (5.10). Among them, X-DER exhibits a higher tolerance to perturbations especially in the high-σ regime, which suggests that its attained minima are overall harder to disrupt when compared to the other methods.

A complementary flatness measure [24, 65, 76] examines the eigenvalues of the Hessian of the overall loss function $\nabla^2_\theta \hat{\mathcal{L}}_{\text{CL}}$, approximated by computing the empirical Fisher Information Matrix on the training set [24, 81]:

$$F \triangleq \sum_{i=0}^{T-1} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{T}_i} \Big[\nabla_\theta \hat{\mathcal{L}}_{\text{CL}}\big(f_\theta(x), y\big) \nabla_\theta \hat{\mathcal{L}}_{\text{CL}}\big(f_\theta(x), y\big)^{\text{T}}\Big]. \tag{5.11}$$

As done in Sec. 4.4.2, we estimate the sum of the eigenvalues of F through the trace of the matrix $\text{Tr}(F)$, reported in Fig. 5.14b. Even according to this metric, DER, DER++ and X-DER reach flatter minima w.r.t. other approaches. Remarkably, X-DER produces lower $\text{Tr}(F)$ values, suggesting that its improved accuracy can be linked to the local geometry of the loss.

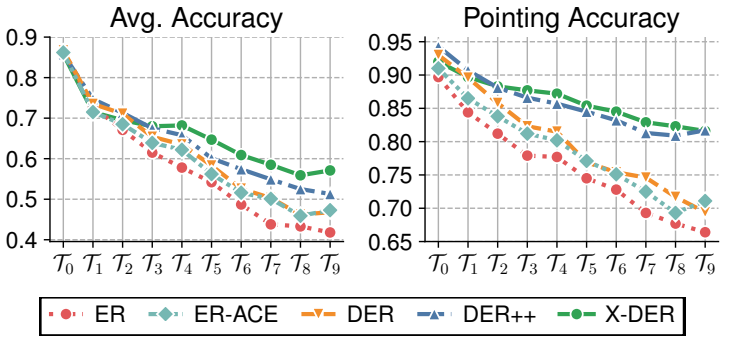


Fig. 5.15: On S-CUB200, the average test-set accuracy (left) and the average pointing accuracy (right).

5.4.5 X-DER Provides Better Explanations for its Predictions

In this section, we present a final analysis aimed at assessing the quality of knowledge acquired by the model. To do so, we focus on ER, DER and X-DER and evaluate their responses through model explanation experiments to get insights into the reasons underlying their predictions.

Model Explanations for Primary Information

Motivated by the investigation carried out in [32], wish to assess the quality of the *visual concepts* encoded in the intermediate layers of the network. More precisely, we are interested in determining whether the use of Knowledge Distillation leads to more refined visual concepts in CL regimes, even with catastrophic forgetting in action.

Since there is no agreement on the exact definition of visual concepts and on how to quantify them given a DNN, we follow [32] and apply the evaluation protocol proposed in [202], called **pointing game**, which characterises the spatial selectiveness of a saliency map in the localisation of target objects. This evaluation consists of the following steps: *i)* given a trained model, take an inference step on a test image and construct an explanation map (*e.g.*, Grad-CAM [150]); *ii)* given the explanation map, check whether the point with the maximum score falls into the object region (usually defined through annotated segmentation maps); *iii)* if it does, we have a *hit* (a *miss* otherwise); *iv)* summarise the results through the average Pointing Accuracy (PA):

$$\text{PA} \triangleq \sum_{y=0}^{T\cdot|\mathcal{Y}|-1} \frac{\#\,\text{hit}_y}{\#\,\text{hit}_y + \#\,\text{miss}_y}. \tag{5.12}$$

PA gives us a compact quantification of whether the CL model ascribes

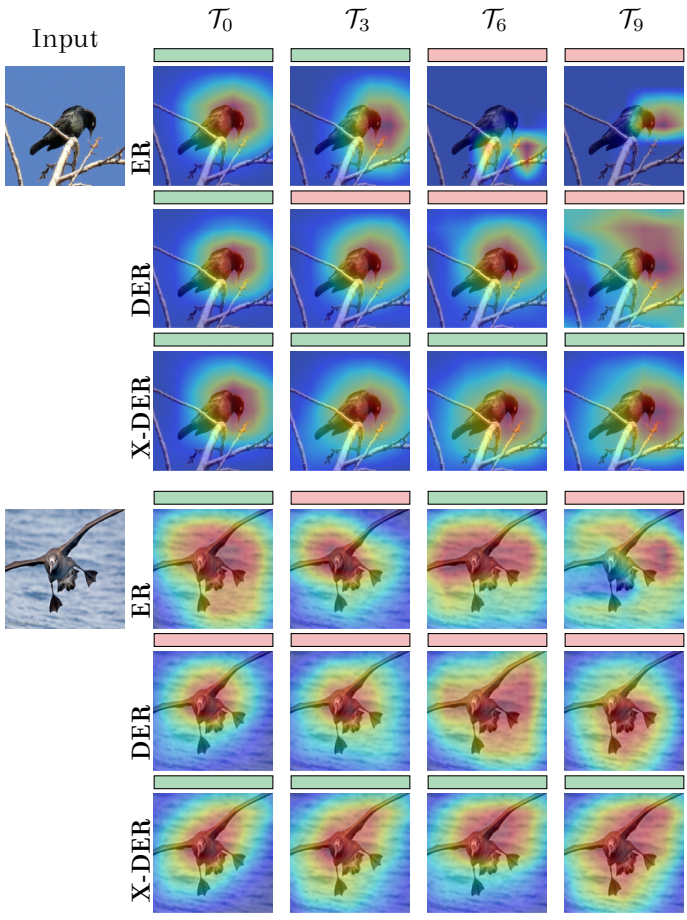


Fig. 5.16: Considering some examples of the first task of S-CUB200, the evolution of explanation maps as tasks progress. Green and red bars indicate whether the model predicts the right class.

its score to the expected spatial locations within the target image, by focusing on the foreground and not the background of the target image.

As the datasets considered so far do not come with segmentation maps, we run this experiment on S-CUB200[5]. On top of that, we extract explanation maps through the Grad-CAM algorithm [150] and use them to compute the resulting PA, which is reported in Fig. 5.15 along with the

[5]It must be noted that, due to the low amount of data in this dataset, we need to resort to a model pre-trained on the ImageNet dataset as done in [27, 32]. We do not follow the exact benchmark settings of Tab. 2.1, but adopt here ResNet-18 as a backbone, train for 70 epochs, use a 0.2× learning rate drop at epochs 20, 40, 60 and optimise our model with RAdam [100] instead of SGD.

Fig. 5.17: Synthetic images obtained by stitching patches of COCO 2017 (Green) examples on top of CIFAR-100 (Red) images.

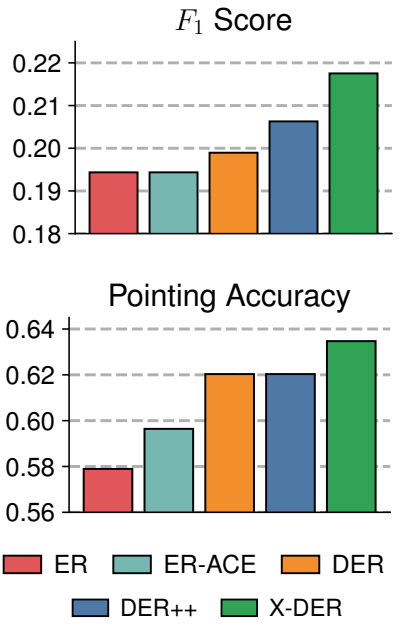


Fig. 5.18: Performance on secondary targets, expressed as F_1 and PA.

incremental classification accuracy. Compared to other approaches, X-DER appears less prone to forgetting the reasoning behind its predictions, as also highlighted by the qualitative examples shown in Fig. 5.16.

Model Explanations for Secondary Targets

After investigating the ability of CL models to motivate their predictions through the right evidence, we now take one more step and devise a similar analysis aimed at measuring whether the same localisation can be successfully applied to secondary information. If we suppose that a test image also contains a *background item* from the problem's training set, we would wish for the models' predictions to capture this secondary information and distribute so as to signal its presence. Furthermore, if the learner can correctly encode the presence of multiple objects in its response, we also expect its activation maps to convey meaningful information for the purpose of their localisation.

To investigate this matter, we evaluate several RBMs by initially training them on S-CIF100. Then, we construct a synthetic benchmark obtained by selecting small image patches from COCO 2017 [97] which

depict classes also present in CIFAR-100 and stitching them over CIFAR-100 images[6]. As shown in Fig. 5.17, the patches are cut through ground-truth segmentation masks and pasted on CIFAR-100 images to simulate secondary semantic content. Finally, we exploit the **linear evaluation** protocol [31] to assess the representation quality of secondary targets: we freeze the backbone network's parameters and only train a linear classifier on top of its features.

We compare the performance of several methods in terms of F_1 score and PA for the *stitched* secondary targets and report the results in Fig. 5.18. We observe that the approaches relying on Knowledge Distillation perform better according to both considered metrics. Notably, X-DER achieves the best metrics, providing a further confirmation of its effectiveness in retaining rich secondary information.

5.5 Conclusions

This chapter started with a preliminary analysis, showing that – while effective – the approach proposed in Chap. 4 discards informative semantic data about the relation between old and novel classes and suffers from a classification bias in favour of recently acquired knowledge.

To address these issues, we proposed X-DER, an extended version of our previous proposal that introduces several innovations addressing the above-mentioned issues. X-DER was first tested with experiments on multiple Class-IL datasets, then further analysed with extensive ablation studies highlighting the reasons for its effectiveness against forgetting.

One of the key insights provided by this chapter, *i.e.*, the effectiveness of preparing the learner for future learning in CL, will also constitute the basis for the RBMs proposed in the next chapters. In particular, Chap. 6 will achieve this goal by modulating the model's capacity so as not to overfit replay data, while Chap. 7 will introduce representation constraints that can be likened to an explicit version of the self-supervised approach taken by X-DER. Beyond the scope of this thesis, we see an increased number of newly proposed RBMs leveraging similar ideas. Most notably, both [133] and [23] propose CL models which dominantly learn through self supervision, highlighting that this drastically reduces the amount of forgetting. Their effectiveness can be justified through the theoretical instruments proposed in this chapter, on the basis that self-supervised features are more easily adapted to future knowledge.

[6] We facilitate the stitching by using a 2x-upscaled version of CIFAR-100 obtained through the CAI super-resolution API [148].

Chapter 6

Modulating Replay Plasticity with Lipschitz Regularisation

6.1 Motivation

The experimental results presented in previous chapters highlight the effectiveness and reliability of RBMs for Class-IL CL. In Sec. 5.1.3, we introduced the classification bias problem that affects many members of this family of methods and specifically mentioned how it might be linked to the repeated optimisation of data stored in a small memory buffer. This aspect was also thoroughly studied in [169], which interprets it as a root cause for overfitting in RBMs.

In this chapter, we expand our analysis of this point, highlighting that the differentiated availability of data from the input stream (*i.e.*, *current-task* data) and replay buffer (*i.e.*, *past* data) produces radically different decision boundaries. As we illustrate in Fig. 6.1, the model's restrained access to only a small portion of past data increases its epistemic uncertainty [75], leading the decision surfaces for past classes to slowly erode everywhere, with the exception of those input regions close to the neighbourhood of buffer data-points (which are repeatedly optimised).

We provide a quantitative evaluation of this phenomenon by following the procedure outlined in [196]. Namely, we take an input example, subject it to perturbations with increasing magnitudes, and track the difference between the correct logit and the maximum incorrect one – the decision boundary is encountered when these two values coincide[1]. Fig. 6.2 visualises this quantity for ER-ACE – chosen here as a robust and well-performing RBM – after each task of S-CIFl0, w.r.t. examples first learnt at $\mathcal{T}_0$ and either included in $\mathcal{M}$ (top row) or left out of it (bottom

[1]Additional details on the construction of this plot are provided in Sec. 6.5.1.

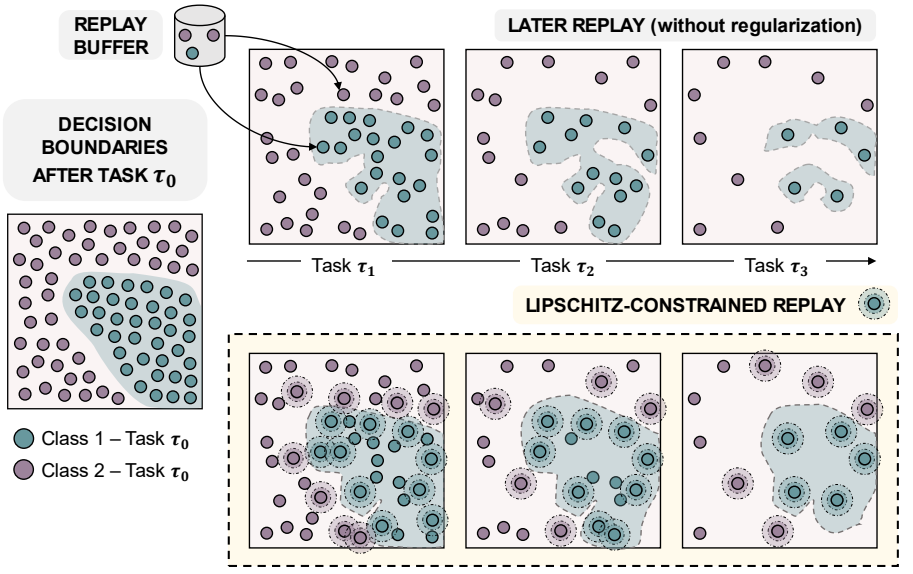


Fig. 6.1: An illustration of the deteriorating decision boundaries of RBMs. Left: the boundary between two classes learnt at task $\mathcal{T}_0$ from the input stream is initially smooth. Right (first row): in subsequent tasks $\mathcal{T}_1 \rightarrow \mathcal{T}_3$, RBMs can access a decreasing number of examples from their replay buffer: the model overfits and the original boundary erodes. By applying our proposed Lipschitz-based learning constraint on replayed data (second row), the model is prevented from excess variation in its responses, avoiding jagged boundaries.

row). We observe a clearly differentiated behaviour: in the former case, boundaries are smooth and robust; in the latter, the region of correct predictions (green) shrinks significantly, indicating that they are easily disrupted.

Driven by this insight, this chapter proposes a novel mechanism for preserving the robustness of decision boundaries in RBMs. We pursue this purpose by bounding the model's Lipschitz constant, which quantifies how the model's response changes in proportion to a change in its input [8]. While this approach has been followed in the adversarial robustness literature [34, 91, 50], its relevance for the case of CL has not yet been investigated. We therefore propose an additional regularisation term called **Lipschitz-Driven Experience Replay (LiDER)**, which can be seamlessly combined with any RBM to counteract boundary deterioration and – as a result – improve its Class-IL performance.

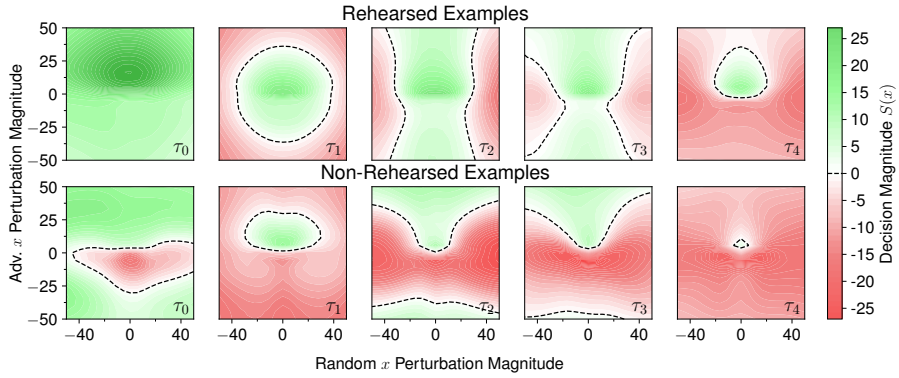


Fig. 6.2: Visualisation of the decision boundaries (dashed curves) for increasing perturbations around data-points from the first task of S-CIF10. As training progresses, points in the memory buffer (top) are subject to much less severe boundary deterioration compared to points from the same classes but not rehearsed (bottom).

6.2 Lipschitz Constant

In this section, we present a brief summary on the mathematics of the Lipschitz constant for DNNs. Subsequently, we propose a simple experiment assessing its relevance in the case of CL.

6.2.1 Lipschitz Constant Computation

A generic function f is said to be *Lipschitz continuous* if there exists a value $L \in \mathbb{R}^+$ such that the following inequality holds:

$$\|f(x) - f(y)\|_2 \leq L \|x - y\|_2, \quad \forall x, y \in \mathbb{R}^n. \tag{6.1}$$

If such a value exists, the smallest L that satisfies the condition is referred to as the Lipschitz constant $\|f\|_L$. Given a single point $x \in \mathbb{R}^n$, we can quantify the Lipschitz constant around x as follows:

$$\|f\|_L^x = \sup_{x \neq y; y \in \mathbb{R}^n} \frac{\|f(x) - f(y)\|_2}{\|x - y\|_2}. \tag{6.2}$$

In the remainder of this section, we will omit the repeated reference to x for the sake of brevity; all following expressions are to be intended as computed around this same input point. Unfortunately, computing the Lipschitz constant of even the simplest multi-layer perceptron is an NP-hard problem [172]. As a result, several works rely on its estimation by means of easy-to-compute upper bounds.

Approximating the Lipschitz Constant of a DNN

We follow the approach proposed in [194, 154] and consider our K-layered feed-forward neural network $h_\theta = (H^K \circ \sigma^K \circ H^{K-1} \circ \sigma^{K-1} \circ \dots H^1)$[2] as a sequence of σ-activated linear functions[3] $H^k : x \mapsto W_k^T x$. By so doing, we can compute the constants of each layer individually and then aggregate them to bound the constant of the entire model. For a given H^k, we have:

$$\left\|H^k\right\|_L = \sup_{x \neq y; y \in \mathbb{R}^n} \frac{\left\|W_k^T x - W_k^T y\right\|_2}{\|x - y\|_2} \overset{\xi = y - x}{=} \sup_{\xi \neq 0; \xi \in \mathbb{R}^n} \frac{\left\|W_k^T \xi\right\|_2}{\|\xi\|_2} = \sigma_{\max}(W_k),$$

where $\sigma_{\max}(W_k)$ is the largest singular value of the weight matrix W_k (also known as its spectral norm $\|W_k\|_{\text{SN}}$). To deal with the non-linear composite functions (*e.g.*, residual blocks) that may appear in our DNN, we leverage the following inequality:

$$\begin{aligned}\left\|g\big(z(x)\big) - g\big(z(y)\big)\right\|_2 &\leq \|g\|_L \|z(x) - z(y)\|_2 \\ &\leq \|g\|_L \|z\|_L \|x - y\|_2 \Rightarrow \|z \circ g\|_L \leq \|g\|_L \|z\|_L ,\end{aligned}$$

where g and z are two Lipschitz-continuous functions characterised by the constants $\|g\|_L$ and $\|z\|_L$. In the case of ReLU-activated networks, the forward pass through σ^k, $k = 1, 2, \dots, K$, can be re-arranged as a matrix multiplication by a diagonal matrix whose elements are either zero or one. Therefore, the corresponding Lipschitz constant $\left\|\sigma^k\right\|_L \leq 1$. With these elements, we can compute an upper bound for the Lipschitz constant of the entire network as follows:

$$\|f_\theta\|_L \leq \left\|H^K\right\|_L \cdot \left\|\sigma^K\right\|_L \cdot \dots \cdot \left\|H^1\right\|_L \leq \prod_{k=1}^{K} \left\|H^k\right\|_L = \prod_{k=1}^{K} \|W_k\|_{\text{SN}}. \tag{6.3}$$

Computing the Spectral Norm of Weight Matrices

The computation of $\|W_k\|_{\text{SN}}$ can generally be accomplished through the Singular Value Decomposition (SVD), which produces – among the others – the largest singular value [115, 50]; however, for complex structures (*e.g.*, convolutions or entire residual blocks) this becomes computationally prohibitive. Hence, we resort to the approximation introduced in [154], which estimates $\|W_k\|_{\text{SN}}$ through the largest eigenvalue λ_1^k of the Transmitting Matrix TM^k:

$$\text{TM}^k \triangleq \left[(F^k)^T (F^{k-1})\right]^T \left[(F^k)^T (F^{k-1})\right], \tag{6.4}$$

where $F^k \in \mathbb{R}^{B \times d_k}$ indicates the L2-normalised feature map produced by the k^{th} layer from a batch of B samples. Finally, we adopt a backpropagation-friendly approach for computing the largest eigenvalue of TM^k via the power iteration method [119].

[2] With h_θ indicating pre-softmax model responses of $f_\theta(\cdot)$, as defined in Sec. 3.3.2.

[3] Other common transformations that make up DNNs (*e.g.*, convolutions, max-pool) can be expressed in terms of matrix multiplications [50].

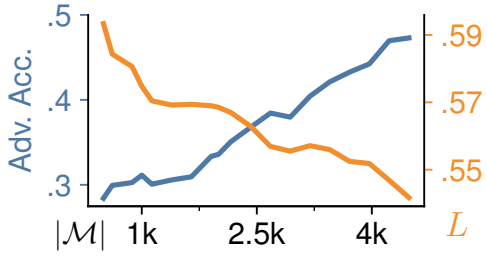


Fig. 6.3: An exemplification of the link between buffer overfitting and the Lipschitz constant in RBMs. For ER-ACE trained on S-CIF100, the accuracy on adversarially perturbed examples increases with $|\mathcal{M}|$ (blue). The inverse trend is shown between $|\mathcal{M}|$ and the Lipschitz constant of the model (orange).

6.2.2 Lipschitz constant in Continual Learning

The Lipschitz constant L of a DNN has been established as a commonplace measure of both smoothness and generalisation [185, 165, 76] and still constitutes a key ingredient for current evaluations of model capacity [14, 51]. Its employment is widespread in adversarial robustness literature, where current approaches pursue its minimisation while learning [92, 167, 91] or are devised so as to have a small constant by design [34, 61]. In other areas, the smoothing effect of L-based regularisation has been favourably applied to both GAN training [115] and neural fields [98].

As these works all operate in the joint *i.i.d.* learning scenario, the relation of Lipschitz regularisation to CL problems has not yet been studied. We here introduce a simple experiment aimed at showing that L can be used to quantify the decision boundary erosion phenomenon in RBMs.

We first train ER-ACE on S-CIF100 several times at different buffer sizes $|\mathcal{M}|$; then, for each trained model, we quantify its ability to withstand a Carlini-Wagner adversarial attack [20] by measuring its resulting accuracy on training-set examples. From the results in Fig. 6.3, we see this accuracy growing with $|\mathcal{M}|$, which aligns with the observations made in the previous section. Indeed, a larger memory buffer makes it harder for the model to overfit its content, which hinders the boundary deterioration effect w.r.t. to the examples of the training set.

Furthermore, we use the approach presented above to estimate the Lipschitz constant L of the model around the same data-points. We observe a clear inversion of the previous trend: without explicit regularisation, the Lipschitz constant of a model increases for smaller memory buffers. In other words, subjecting the model to a low replay-data training regime leads to a function space highly sensitive w.r.t. input perturbations.

6.3 Lipschitz-Driven Experience Replay

We propose to apply Lipschitz-based regularisation to the continual learner to mitigate overfitting on buffer data-points. To achieve this, we require each layer k in the backbone to behave as a c_k-Lipschitz continuous function, for a given real positive target c_k:

$$\mathcal{L}_{\text{c-Lip}} \triangleq \sum_k |\lambda_1^k - c_k|, \tag{6.5}$$

where λ_1^k is computed as the largest eigenvalue of TM^k (in line with Eq. 6.4) using only the activation maps of replayed exemplars.

The target constants c_k, could in principle be regarded as hyper-parameters of our learning objective (as done in [98]); however, this would imply fixing an *a-priori* budget for each layer's amount of flexibility, which is complex especially a CL scenario where there is no access to the full data distribution. Instead, we empirically observe (see Sec. 6.5.4) that it is more beneficial to let the c_ks be optimised by means of gradient descent, interpreting them as additional learnable parameters that represent the appropriate level of sensitivity that should be enforced for each layer. However, since this approach may produce trivial solutions by maximising c_k, we need to introduce a second learning objective aimed at keeping the estimated upper bounds close to zero:

$$\mathcal{L}_{\text{0-Lip}} \triangleq \sum_k |\lambda_1^k|. \tag{6.6}$$

Intuitively, when $\lambda_1^k \to 0$, the outputs of the corresponding k^{th} layer have low sensitivity to changes in its input. This can be seen as imposing a limit on the capacity of the hypothesis class subsuming the model [51], effectively requiring it to behave *as if* underparameterised. By so doing, the model cannot afford to learn a jagged decision surface, thus limiting the decision surface erosion phenomenon that we observed on rehearsal classes in Sec. 6.1.

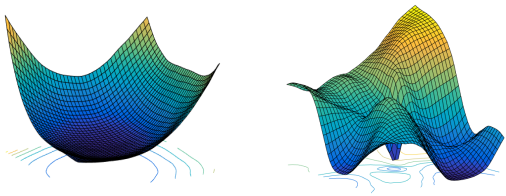
We now formulate the overall objective of **Lipschitz-Driven Experience Replay (LiDER)** by combining the two introduced loss terms; formally:

$$\mathcal{L}_{\text{LiDER}} \triangleq \alpha\mathcal{L}_{\text{c-Lip}} + \beta\mathcal{L}_{\text{0-Lip}}. \tag{6.7}$$

This objective can be easily combined with existing RBMs by adding it to their respective $\mathcal{L}_{\text{R}}$ in Eq. 2.2. This addition produces minimal computation overhead and requires no alteration of the sampling technique used to construct $\mathcal{M}$.

Relation with other regularisation approaches

At first glance, the regularisation of our approach could be understood as a mean to enforce flat minima for each of the tasks, as advocated by



(a) ResNet's loss surface in parameter space*

(b) ResNet's loss surface in input space*

* The magnitude of the two surfaces indicates the loss value, while the left surfaces changes with neural network parameters (e.g. weights and biases update), and the right surface changes with input variation (e.g. noises on test images).

Fig. 6.4: Image from *Yu et al.* [196]. (a) Loss Surface of ResNet in Parameter Space. (b) Loss Surface of ResNet in the Input Space. The latter loss surface demonstrates significant non-smooth variation even though the former does not.

Mirzadeh et al. [113] and *Yin et al.* [192]. However, these approaches operate in the **parameter space** and pursue flatness of the loss landscape w.r.t. weights, *i.e.*, they encourage the model to be robust when perturbations are applied to its **weights**. Differently, LiDER aims at achieving robustness w.r.t. changes in **input space**. Even though they may exploit the same mathematical tools – the Hessian and Lipschitz continuity – the two strategies build upon orthogonal axes (weights *vs* input).

In this respect, the relation between these is not clearly understood and worth exploring [190, 164, 196, 71]. The authors of [190] report that there exists no theoretical correlation between the Hessian w.r.t. weights and the robustness of the model w.r.t. the input. Such a statement is corroborated by Fig. 1 of [196], which we report in Fig. 6.4: although a flat minimum is reached in parameter space, non-smooth variations appear in input space. However, the authors of [190] empirically find that models with higher Hessian spectrum w.r.t. weights are also more prone to adversarial attacks. A similar thesis has been argued by the authors of [196], while the third result reported in [71] seems to refute it. In Sec. 6.5.1, we investigate the opposite link and reveal that RBMs trained to be robust w.r.t. input changes tend to attain flatter minima in parameter space.

6.4 Experiments

In this section, we evaluate our proposed LiDER in the Class-IL setting by combining it with five high-performance RBMs: iCaRL, DER++, X-DER $^{\text{RPC}}_{\text{future}}$, GDumb and ER-ACE. As usual, FT and JT are reported as a lower and upper bound. Results are provided as FAA and FAF on S-

FAA (FAF)	S-CIF100			
Pre-training	–		TinyImgNet	
JT	73.29 (–)		75.20 (–)	
FT	09.29 (86.62)		09.52 (92.31)	
$\lvert\mathcal{M}\rvert$	500	2000	500	2000
iCaRL	44.04 (21.70)	50.23 (17.92)	56.00 (19.27)	58.10 (16.89)
+ **LiDER**	47.02 (21.89)	51.21 (17.13)	57.24 (19.16)	60.97 (15.49)
DER++	37.13 (49.80)	52.08 (31.10)	43.65 (48.72)	58.05 (29.65)
+ **LiDER**	39.25 (45.50)	53.27 (27.51)	45.37 (48.16)	60.88 (25.16)
X-DER $^{\text{RPC}}_{\text{future}}$	44.62 (31.84)	54.44 (17.01)	57.45 (16.86)	62.46 (12.07)
+ **LiDER**	45.22 (28.38)	54.71 (11.33)	57.76 (15.98)	62.78 (11.26)
GDumb	09.28 (–)	19.69 (–)	23.09 (–)	36.05 (–)
+ **LiDER**	10.22 (–)	26.15 (–)	26.09 (–)	41.98 (–)
ER-ACE	36.48 (38.21)	48.41 (27.90)	48.19 (31.84)	57.34 (25.48)
+ **LiDER**	38.43 (36.00)	50.32 (28.30)	48.97 (28.58)	57.39 (25.37)

Tab. 6.1: For different RBMs, Class-IL FAA and FAF on S-CIF100 with and without LiDER.

CIF100, S-*mini*Img and S-CUB200[4] and entail both from-scratch training and pre-trained initialisation of the backbone model; for this latter case, we report the results on S-CIF100 with pre-training on a resized 32×32 version of Tiny ImageNet and on S-CUB200 with pre-training on ImageNet. This scenario is particularly interesting for two reasons: *i)* as shown in [110], pre-training implicitly mitigates forgetting by widening the local minima found in function space, thus making the model more robust to input perturbations; *ii)* it has clear practical implications for real-world scenarios where pre-training is typically applied.

6.4.1 Comparison with RBMs

We report the results of our evaluation in Tab. 6.1 and 6.2; LiDER improves the performance of all base methods in all evaluated scenarios in terms of FAA and almost always in terms of FAF[5]. Most notably, it leads to a consistent performance increase in methods such as DER++ and iCaRL, which already feature compelling results, suggesting that their higher generalisation capability can still benefit from increased decision boundary smoothness. GDumb with small $\lvert\mathcal{M}\rvert$ dramatically fails to prevent forgetting and benefits less from LiDER, while still obtaining an

[4]We diverge from Tab. 2.1 by adopting a batch size of 64 for S-CIF100 and of 16 for S-CUB200.

[5]We remark that a slight decrease in FAF might be linked to improved FAA, as mentioned in Sec. 2.4.

FAA (FAF)	**S-*mini*Img**		**S-CUB200**	
Pre-training	–		ImgNet	
JT	53.55 (–)		78.54 (–)	
FT	04.51 (77.38)		08.56 (82.38)	
$\lvert\mathcal{M}\rvert$	2000	5000	400	1000
iCaRL	22.58 (16.46)	22.78 (16.37)	56.52 (13.43)	60.09 (11.41)
+ **LiDER**	23.22 (11.21)	23.95 (11.18)	57.12 (14.31)	60.37 (10.89)
DER++	23.44 (46.69)	30.43 (37.11)	49.30 (36.05)	61.42 (19.95)
+ **LiDER**	28.33 (36.29)	35.04 (25.02)	57.90 (27.55)	67.97 (14.44)
X-DER $^{\text{RPC}}_{\text{future}}$	26.38 (38.33)	29.91 (28.29)	58.23 (16.58)	64.90 (09.03)
+ **LiDER**	29.15 (27.18)	32.56 (20.59)	60.00 (15.64)	65.98 (08.64)
GDumb	15.22 (–)	27.79 (–)	09.36 (–)	18.98 (–)
+ **LiDER**	15.24 (–)	29.49 (–)	09.67 (–)	19.51 (–)
ER-ACE	22.60 (23.74)	27.92 (19.72)	41.83 (26.42)	51.98 (18.79)
+ **LiDER**	24.13 (25.97)	30.00 (19.99)	50.89 (20.79)	60.92 (14.62)

Tab. 6.2: For different RBMs, Class-IL FAA and FAF on S-*mini*Img and S-CUB200 with and without LiDER.

improvement. However, upon increasing its buffer size and/or providing pre-trained initialisation, introducing LiDER determines a considerable performance increase (on S-CIF100, a FAA gain of +0.98% for $|\mathcal{M}| = 500$ grows to +6.46% for $|\mathcal{M}| = 2000$ and to +2.99% if pre-train is added).

On average, LiDER produces a FAA gain of 2.32%, 2.08% and 4.36% on S-CIF100, S-*mini*Img, and S-CUB200 respectively.

6.4.2 Comparison with Regularisation Approaches

To provide a thorough evaluation, we further compare our proposal with three existing regularisation techniques: sSGD, oEWC and OLAP. The former approach alters the training regime to bias the optimisation towards flat minima in the loss landscape; the latter two constrain the most important parameters for old tasks to remain close to their past values. While these approaches can in principle be employed on their own to prevent catastrophic forgetting (albeit with mixed results on Class-IL), we here combine them with ER-ACE and DER++ to ascertain whether they can benefit RBMs as additional training objectives, like LiDER.

As reported in Tab. 6.3, sSGD boosts ER-ACE and DER++ only on S-CIF100; on the contrary, its performance degrades severely both on the more complex S-*mini*Img and on S-CUB200, where it appears to fail to effectively exploit the pre-trained network. By contrast, we find the application of oEWC and OLAP to be rewarding, especially in the presence of pre-training. In this respect, we recall that pre-training has a known

FAA	S-CIF100		S-*mini*Img		S-CUB200	
Pre-training	–		–		ImgNet	
$\lvert\mathcal{M}\rvert$	500	2000	2000	5000	400	1000
ER-ACE	36.48	48.41	22.60	27.92	41.83	51.98
+ sSGD	**39.59**	49.70	22.43	24.12	22.67	29.88
+ oEWC	35.06	45.59	**24.32**	29.46	48.34	59.74
+ OLAP	36.58	47.66	23.19	28.77	42.64	52.86
+ **LiDER**	38.43	**50.32**	24.13	**30.00**	**50.89**	**60.92**
DER++	37.13	52.08	23.44	30.43	49.30	61.42
+ sSGD	38.48	50.74	19.29	24.24	31.08	41.69
+ oEWC	35.22	51.53	24.53	31.91	51.86	62.54
+ OLAP	34.48	50.80	25.02	32.78	49.56	63.27
+ **LiDER**	**39.25**	**53.27**	**28.33**	**35.04**	**57.90**	**67.97**

Tab. 6.3: Class-IL FAA comparison between different regularisation strategies applied on top of ER-ACE and DER++.

flattening effect on the loss landscape [110], which makes encouraging the model to stay close to its prior particularly beneficial. All things considered, LiDER proves almost always more effective than any of the other tested approaches, validating our approach aimed at regularising model responses.

6.5 Analysis

In this section, we present several analytical experiments aimed at characterising the effects produced by LiDER on the model and at stressing the limits and design choices of the proposed approach.

6.5.1 Effects on generalisation

Decision surface of LiDER

Fig. 6.2 depicts the model's tolerance to input perturbations in the form of a decision surface plot [196]. This visualisation is constructed by focusing on a set of perturbations $x_p \triangleq x + i \cdot \alpha + j \cdot \beta$ computed around a data-point x, with α being a random divergence direction and β corresponding to the direction induced by the first step of a non-targeted Fast Gradient Signed Method attack [85]. The plot shows the respective values of the decision function $S(x_p)$, where $S(x) \triangleq h_\theta(x)_t - \max_{i \neq t} h_\theta(x)_i$, and highlights decision boundary of the model (*i.e.*, the locus of $\{x_p; S(x_p) = 0\}$), in correspondence of which the model fails its prediction.

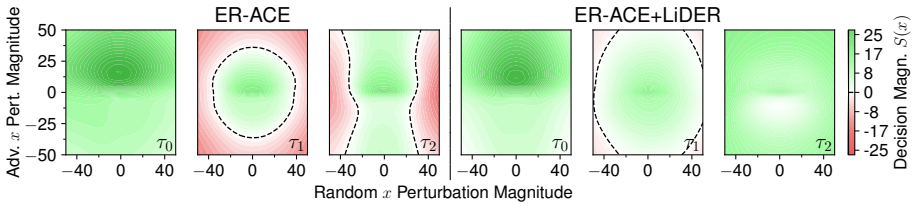


Fig. 6.5: Effect of LiDER on the robustness of the decision boundary produced by ER-ACE across subsequent tasks. Same setup as Fig. 6.2.

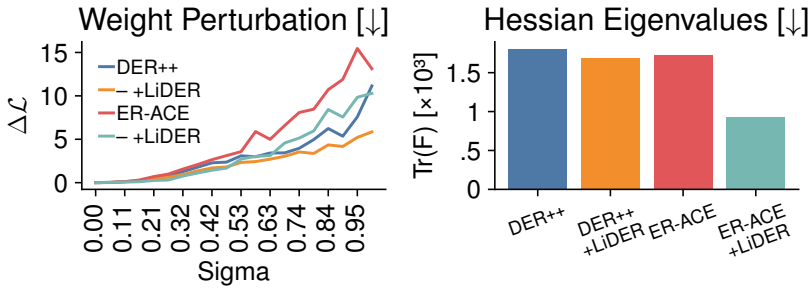


Fig. 6.6: (Left) Robustness of models regularised with LiDER against weight perturbations. (Right) Flatness around the minimum found during optimisation, measured as sum of the eigenvalues of the Hessian matrix.

In Fig. 6.5, we adopt the same approach to compare the decision boundaries around rehearsed $\mathcal{T}_0$ examples for ER-ACE with and without LiDER. While both models start with a similarly robust decision landscape in $\mathcal{T}_0$, later tasks reveal a clear shrinking behaviour in ER-ACE. On the contrary, introducing L-based regularisation leads to minimal decision boundary deterioration in later tasks.

Loss Landscape of LiDER

In the previous chapters, we investigated the generalisation capabilities of CL models by evaluating the flatness of their attained minima. Here we do the same with LiDER, reporting the results of the two evaluations of Sec. 4.4.1 and 5.4.4 in Fig. 6.6. We see that DER++ and ER-ACE combined with LiDER both improve their resilience to weight perturbations and achieve lower Hessian eigenvalues. This is in line with the common interpretation of the Lipschitz constant of a DNN as a measure of generalisation capabilities [185, 165].

p	DER++	+LiDER
.0%	37.13	**39.25**
.01%	36.13	**38.08**
.1%	31.35	**35.53**
.25%	28.74	**30.78**

Tab. 6.4: Class-IL FAA with buffer poisoning for DER++ with and without LiDER.

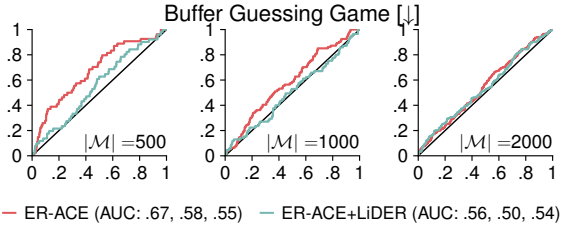


Fig. 6.7: ROC curves for the Buffer Guessing Game, showing the likelihood of a given sample belonging to $\mathcal{M}$.

6.5.2 Effects on $\mathcal{M}$ in realistic scenarios

In this section, we investigate the implications of adopting a LiDER-enhanced RBM in realistic evaluation settings. We propose two new experiments that focus on the behaviour of the memory buffer, namely on its ability to handle incorrect labels and on its privacy.

Robustness to buffer poisoning

As a first study, we acknowledge that a real-world CL application might need to deal with incorrect annotation of the input data-points. While typically overlooked in standardised benchmarks, this aspect is particularly critical when working with RBMs. As previously shown, samples included in the buffer are most likely to be overfit, which would induce a severe loss of performance in the case of an incorrect label.

We assess this effect through *label poisoning*: while training DER++ on S-CIF100 ($|\mathcal{M}| = 500$), items sampled for inclusion in $\mathcal{M}$ are randomly assigned a wrong label from the current task with probability p. Tab. 6.4 reports the resulting FAA, showing an expected performance degradation as p increases. We observe that the application of LiDER allows for achieving a higher accuracy even in the case of poisoning, confirming our intuition that our proposal alleviates overfitting of elements stored in the memory buffer.

Buffer Guessing game

To further illustrate the implication of RBMs overfitting in realistic CL scenarios, we propose a simple experiment called *buffer guessing game*. As we posit that $\mathcal{M}$ plays a much larger role in shaping the decision boundary w.r.t. the input stream, we take ER-ACE fully trained on a S-CIF100 and – using the full training set $\mathcal{X}_0$ of $\mathcal{T}_0$ – we try to find $\mathcal{M} \cap \mathcal{X}_0$ (*i.e.*, the subset of data-points that are included in the model's buffer).

FAA	**S-CIF100**				**S-*m*Img**		**S-CUB200**	
Pre-training	–		TinyImgNet		–		ImgNet	
$\lvert\mathcal{M}\rvert$	500	2000	500	2000	2000	5000	400	1000
ER-ACE	36.48	48.41	48.19	57.34	22.60	27.92	41.83	51.98
+ LiDER **(curr. task)**	37.54	**50.37**	48.94	57.07	23.35	29.25	48.44	59.60
+ LiDER **(buffer)**	**38.43**	50.32	**48.97**	**57.39**	**24.13**	**30.00**	**50.89**	**60.92**
DER++	37.13	52.08	43.65	58.05	23.44	30.43	49.30	61.42
+ LiDER **(curr. task)**	34.78	49.76	44.48	59.39	24.84	31.05	56.96	66.63
+ LiDER **(buffer)**	**39.25**	**53.27**	**45.37**	**60.88**	**28.33**	**35.04**	**57.90**	**67.97**

Tab. 6.5: Class-IL FAA when regularising over examples from the current task (stream) or buffer data-points (standard LiDER).

We do so by attaching to each $x \in \mathcal{T}_0$ a score s_x computed in a neighbourhood of x. s_x quantifies the mean *height* of the decision surface, *i.e.*, the difference between the predicted probability of the right class and the one of the highest wrong class. As in [196], we model the neighbourhood by leveraging random perturbations; moreover, we compute s_x w.r.t. to the Task-IL prediction function in order to avoid the influence of inter-task biases on our results. Finally, we measure our ability to identify in-buffer examples by calculating the ROC curve obtained from these scores. Fig. 6.7 reports the results of this experiment at different buffer sizes. We see that: *i)* ER-ACE makes it easier to reconstruct the content of the buffer, as indicated by larger ROC-AUC scores w.r.t. ER-ACE+LiDER; *ii)* in line with our expectations, this effect is increased when employing smaller memory buffers, as this leads to the repeated optimisation of a smaller pool of data.

6.5.3 Applying LiDER on current-task examples

While our initial study on buffer overfitting led us to apply LiDER exclusively on buffer data-points, no technical reason prevents its application on current-task data. In Tab. 6.5 we report the results obtained by switching the regularisation target to the data from the input stream: we observe that doing so produces worse FAA results.

This simple ablation aligns with our initial intuition, suggesting that – thanks to the abundance of current-task data available – the model's epistemic uncertainty [75] on current classes is low and the learnt decision boundaries are likely to be smooth. In this case, introducing an additional Lipschitz regularisation term produces too much rigidity and restrains the learning with no advantages.

FAA	S-CIF100			
Pre-training	–		TinyImgNet	
$\lvert\mathcal{M}\rvert$	500	2000	500	2000
DER++ + LiDER Fixed Targets	36.42	51.52	43.16	59.53
DER++ + LiDER Eq. 6.7	**39.25**	**53.27**	**45.37**	**60.68**
ER-ACE + LiDER Fixed Targets	34.99	46.70	45.21	54.82
ER-ACE + LiDER Eq. 6.7	**38.43**	**48.97**	**48.97**	**57.39**

Tab. 6.6: Class-IL FAA when adopting fixed Lipschitz targets c_k and learned targets (standard LiDER).

6.5.4 Optimisation with a fixed target

As we mentioned when first introducing Eq. 6.5, the target Lipschitz values c_k could be fixed prior to training as done in [98], thus avoiding the need for Eq. 6.6. In Tab. 6.6 we empirically show that doing so does not lead to satisfactory results in a CL setting, with our proposal consistently outperforming the fixed-target approach.

6.6 Conclusions

In this chapter, we illustrated the existence of a differentiated learning regime affecting input-stream and rehearsal classes in RBMs. This disparity inevitably leads to memory buffer overfitting, which is a known Achilles' heel for this class of CL methods [169]. To deal with this effect, we introduced a plug-in regularisation term called LiDER, which bounds the complexity of the in-training model at replay time. We highlighted that our proposal gives a consistent performance boost when combined with SOTA RBMs and highlighted its effect on the model by means of additional analysis.

The basic intuition to introduce functional regularisation on the model's backbone will also be explored in the next chapter, leading to another plug-in loss term for RBMs that enforces specific geometric characteristics in their latent space. This chapter also introduced some experiments leveraging a pre-trained continual learner, showing that they generally result in higher metrics at the end of training. The significance and implications of such practice will be the focus of Chap. 10.

Among the aspects explored in this chapter, the preliminary experiment on the handling of unreliable labelling appears to be particularly promising as a future research direction, with a series of very recent works starting to explore the topic of CL with noisy labels [79, 12, 73]. In

light of the improved resilience of LiDER to incorrect training targets, there is reason to believe that the analysis of the Lipschitz constant of the online learner might provide useful insights for facilitating the detection of inconsistently labelled exemplars.

Chapter 7

Latent Space Modelling via Geometric Constraints

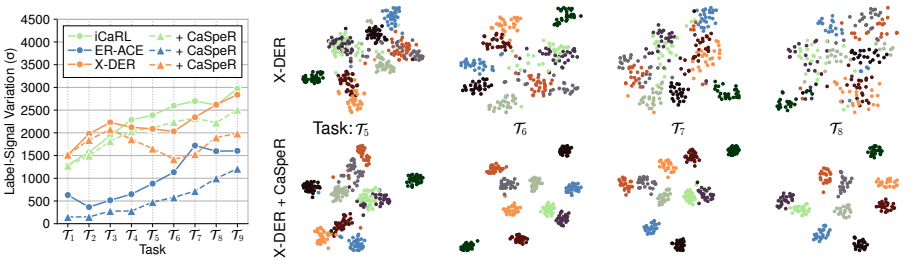
7.1 Motivation

Both Chap. 5 and 6 delved into factors hindering the acquisition of knowledge from the replay buffer in RBMs: namely, the difference in gradient magnitude between the input stream and $\mathcal{M}$ (Sec. 5.1.3) and the deterioration of the model's decision surface due to overfitting (Sec. 6.1). This chapter discusses and addresses a third issue of replay, focusing specifically on changes occurring in its latent space as tasks progress.

We observe that the learner struggles to separate latent projections of replay examples belonging to different classes, making the downstream classifier prone to interference whenever the input distribution changes and representations are perturbed. Motivated by the Riemannian nature of the latent space of DNNs [10], we study this issue by leveraging the toolset of spectral geometry, which favourably allows manipulating the structure of network representations as a whole without imposing constraints on individual coordinates.

To understand how latent space changes in response to the introduction of a novel task on the input stream, we analyse the Latent Geometry Graph (LGG) $\mathcal{G}$ after training on $\mathcal{T}_1, \dots, \mathcal{T}_{T-1}$. $\mathcal{G}$ is constructed by taking all replay examples, forwarding them through the model to obtain the corresponding set of pre-classifier features $\{g \triangleq h_\theta^{\text{pre-clf}}(x); x \in \mathcal{M}\}$ and finally building a k-NN graph on top of it [86]. A compact measure of latent space sparsity w.r.t. classes representations is given by the Label-Signal Variation σ [86] on the adjacency matrix $A \in \mathbb{R}^{|\mathcal{M}| \times |\mathcal{M}|}$ of $\mathcal{G}$:

$$\sigma \triangleq \sum_{i=1}^{|\mathcal{M}|} \sum_{j=1}^{|\mathcal{M}|} \mathbb{1}_{y_i^b = y_j^b} a_{i,j} \,, \tag{7.1}$$



(a) (b)

Fig. 7.1: Illustrations of the alterations occurring in RBMs' latent spaces. (a) A quantitative evaluation measured as Label-Signal Variation (σ) within the LGG for buffer data-points – *lower is better*; (b) t-SNE embedding of the features computed by X-DER $^{\text{RPC}}_{\text{future}}$ for buffered examples in later tasks (top). Interference between classes is visibly reduced if CaSpeR is applied (bottom). All experiments are carried out on S-CIFl00, (a) has $|\mathcal{M}| = 500$, (b) has $|\mathcal{M}| = 2000$.

where $\mathbb{1}_{\cdot}$ is the indicator function and $a_{i,j}$ denotes the $(i,j)^{\text{th}}$ element of A, which is 1 if the i^{th} element of $\mathcal{M}$ is in the k-NN set of the j^{th} and 0 otherwise. We evaluate this metric for three SOTA RBMs and report the results in Fig. 7.1a. We observe that they exhibit a steadily growing σ, indicating that examples from distinct classes are increasingly entangled in later tasks. This effect can also be observed qualitatively by considering a t-SNE embedding of the points in $\mathcal{M}$ (shown in Fig. 7.1b for X-DER $^{\text{RPC}}_{\text{future}}$), which also reveals decreasing distances between examples from different classes as training progresses.

In light of these observations, we introduce a novel loss term aimed at ensuring a cohesive structure in the latent space of RBMs. Our proposed approach, called **Continual Spectral Regulariser (CaSpeR)** (illustrated in Fig. 7.2), leverages graph-spectral theory to promote the generation of well-separated latent embeddings. As anticipated by the results in Fig. 7.1, it can be effectively combined with existing RBMs to improve classification accuracy and robustness against catastrophic forgetting.

7.2 Continual Spectral Regulariser

Our method builds upon the intuition that the latent spaces of DNNs bear a structure informative of the data space they are trained on [155]. By applying a geometric regularisation term, we seek to enforce a desirable structure for latent representations, *i.e.*, partitioning the vertices of $\mathcal{G}$ into well-separated subgraphs with high internal connectivity.

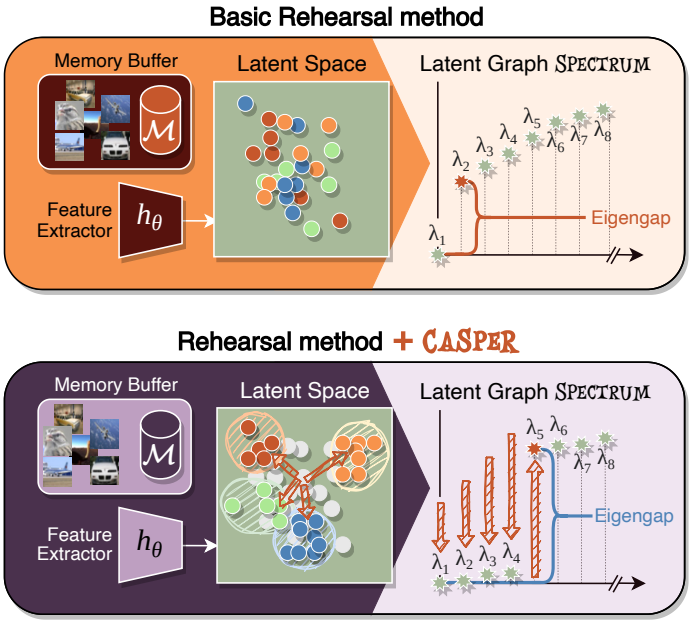


Fig. 7.2: An overview of the proposed CaSpeR regulariser. RBMs struggle to separate the latent-space projections of replay data-points. Our proposal targets the spectrum of the latent geometry graph to induce a partitioning behaviour by maximising the *eigengap* for the number of seen classes.

After computing $\mathcal{G}$ over $\mathcal{M}$ as specified in the previous section, we consider its adjacency matrix A, calculate its degree matrix D and then compute its normalised Laplacian L as:

$$L = I - D^{-1/2} A D^{-1/2}\,, \tag{7.2}$$

where I is the identity matrix. Our regularisation term insists on the eigenvalues λ of L, which we compute and sort by ascending order. Let $c_{\mathcal{M}}$ be the number of different classes within the buffer, we calculate the **Continual Spectral Regulariser (CaSpeR)** regularising loss as:

$$\mathcal{L}_{\text{CaSpeR}} \triangleq -\lambda_{c_{\mathcal{M}}+1} + \sum_{j=1}^{c_{\mathcal{M}}} \lambda_j\,. \tag{7.3}$$

The proposed loss term is weighted through the hyper-parameter ρ and added to the base RBM loss of Eq. 2.2. Through Eq. 7.3, we increase the eigengap $\lambda_{c_{\mathcal{M}}+1} - \lambda_{c_{\mathcal{M}}}$ while minimising the first $c_{\mathcal{M}}$ eigenvalues. A body of results from spectral graph theory, dating back at least to [29, 159, 156], explain the gap occurring between neighbouring Laplacian eigenvalues

Alg. 7.1: CaSpeR Loss Computation

Input: Memory buffer $\mathcal{M}$ of saved samples
$x^b \leftarrow \text{BalancedSampling}(\mathcal{M})$
$g^b \leftarrow h_\theta^{\text{pre-clf}}(x^b)$
$A \leftarrow \text{k-NN}(g^b)$
$D \leftarrow \text{diag}(\sum_i a_{1,i}, \sum_j a_{2,i}, ..., \sum_i a_{b,i})$
$L \leftarrow I - D^{-1/2} A D^{-1/2}$ ▷ Eq. 7.2
$\lambda \leftarrow \text{Eigenvalues}(L)$
$\mathcal{L}_{\text{CaSpeR}} \leftarrow -\lambda_{c_\mathcal{M}+1} + \sum_{j=1}^{c_\mathcal{M}} \lambda_j$ ▷ Eq. 7.3

as a quantitative measure of graph partitioning. Our proposal draws on these results but turns the *forward* problem of computing the optimal partitioning of a given graph, into the *inverse* problem of seeking a graph with the desired partitioning. Following the intuition that the number of eigenvalues close to zero corresponds to the number of loosely connected partitions within the graph [90], our loss indirectly encourages the points in the buffer to be clustered without strict supervision. A step-by-step summary of the outlined procedure can be found in Alg. 7.1[1].

Efficient Batch Operation

While seemingly straightforward, the operation of CaSpeR entails the cumbersome task of constructing the entire LGG $\mathcal{G}$ at each forward step. Indeed, accurately mapping the model's ever-changing latent space requires processing all available replay examples in $\mathcal{M}$, which is typically orders of magnitude larger than a batch of examples on the input stream.

To avoid a slow training procedure with high memory requirements, we propose an efficient approximation of our initial objective. Instead of operating on $\mathcal{G}$ directly, we sample a randomly chosen sub-graph $\mathcal{G}_\text{p} \subset \mathcal{G}$ spanning only p out of the $c_\mathcal{M}$ classes represented in the memory buffer. As $\mathcal{G}_\text{p}$ still includes a conspicuous number of nodes, we further sub-sample and extract $\mathcal{G}_\text{p}^t \subset \mathcal{G}_\text{p}$, a smaller graph with t exemplars for each class.

By repeating these random samplings in each forward step, we optimise a Monte Carlo approximation of Eq. 7.3:

$$\mathcal{L}^*_{\text{CaSpeR}} \triangleq \mathop{\mathbb{E}}_{\mathcal{G}_\text{p} \subset \mathcal{G}} \left[\mathop{\mathbb{E}}_{\mathcal{G}_\text{p}^t \subset \mathcal{G}_\text{p}} \left[-\lambda_{p+1}^{\mathcal{G}_\text{p}^t} + \sum_{j=1}^{p} \lambda_j^{\mathcal{G}_\text{p}^t} \right] \right], \tag{7.4}$$

where the $\lambda^{\mathcal{G}_\text{p}^t}$ denote the eigenvalues of the Laplacian of $\mathcal{G}_\text{p}^t$. It must be noted that we enforce the eigengap at p, as we know by construction that each $\mathcal{G}_\text{p}^t$ comprises samples from p communities within $\mathcal{G}$.

[1] Since our proposal relies on the availability in $\mathcal{M}$ of a minimum number of samples for each class, we adopt BRS as proposed in Sec. 3.1.

FAA (FAAF)	**S-CIF100**			
Method	**Class-IL**		**Task-IL**	
JT	63.11 (–)		88.81 (–)	
FT	8.38 (100.00)		30.10 (62.84)	
$\lvert\mathcal{M}\rvert$	500	2000	500	2000
ER-ACE	34.99 (51.41)	46.63 (28.78)	73.86 (10.73)	80.69 (5.37)
+ **CaSpeR**	36.70 (46.61)	47.85 (27.73)	75.14 (4.91)	81.57 (4.93)
iCaRL	39.80 (32.73)	40.54 (32.61)	78.38 (5.38)	78.47 (4.91)
+ **CaSpeR**	40.57 (32.31)	41.83 (25.55)	79.31 (4.61)	79.43 (3.41)
DER++	28.01 (57.56)	42.27 (34.94)	70.55 (11.12)	78.60 (5.96)
+ **CaSpeR**	32.16 (53.41)	46.34 (30.08)	73.25 (9.49)	80.78 (3.04)
X-DER $^{\text{RPC}}_{\text{future}}$	35.89 (44.54)	46.37 (23.57)	77.28 (2.43)	82.55 (0.92)
+ **CaSpeR**	38.23 (43.90)	50.39 (17.65)	78.26 (5.47)	83.77 (0.27)
PODNet	28.16 (58.49)	32.12 (46.73)	67.37 (19.76)	69.63 (15.16)
+ **CaSpeR**	31.40 (48.50)	36.97 (39.00)	70.81 (15.26)	71.90 (11.32)

Tab. 7.1: FAA (FAAF) on S-CIF100 for RBMs with and w/o CaSpeR.

7.3 Experiments

We evaluate CaSpeR both in the Class-IL and Task-IL settings by applying it on top of five SOTA RBMs: ER-ACE, iCaRL, DER++, X-DER $^{\text{RPC}}_{\text{future}}$ and PODNet. Evaluation is carried out on S-CIF100 and S-*mini*Img[2], with results reported in Tab. 7.1 and 7.2 respectively in terms of FAA and FAAF. At a first glance, we observe that CaSpeR leads to a steady improvement in FAA across all evaluated methods and settings. However, some interesting additional trends emerge upon closer examination.

Firstly, we notice that the improvement in accuracy does not grow with the memory buffer size. This is in contrast with the typical behaviour of replay regularisation terms [23, 28]. We believe such a tendency to be the result of our distinctively geometric approach: as spectral properties of graphs are understood to be robust w.r.t. to coarsening [68], CaSpeR does not need a large pool of data to be effective.

Remarkably, the majority of the evaluated methods achieve comparable FAA gains for both CL settings on S-CIF100; this suggests that our method allows the model to better learn and consolidate each task individually (Task-IL) while providing balanced responses for both stream and replay classes (Class-IL). This second tendency is further confirmed by the conspicuous reduction in Class-IL FAAF, which indicates that CaSpeR

[2]W.r.t. Tab. 2.1, we train for 20 epochs with no lr scheduling and use batch size 64 for S-CIF100; we train for 50 epochs with a 0.1 decay factor applied to lr at epochs 35 and 45 using batch size 64 for S-*mini*Img.

FAA (FAAF)	S-*mini*Img			
Method	**Class-IL**		**Task-IL**	
JT	52.76 (–)		87.39 (–)	
FT	3.87 (100.00)		24.05 (67.37)	
$\lvert\mathcal{M}\rvert$	2000	5000	2000	5000
ER-ACE	22.03 (49.04)	27.26 (29.99)	69.05 (13.72)	72.78 (8.93)
+ **CaSpeR**	23.36 (47.90)	29.15 (28.36)	69.59 (13.05)	74.14 (8.12)
iCaRL	19.42 (36.89)	20.17 (33.23)	70.35 (3.92)	70.44 (2.68)
+ **CaSpeR**	20.46 (35.90)	21.45 (32.26)	71.19 (3.67)	71.93 (3.65)
DER++	20.88 (74.48)	28.55 (61.03)	69.78 (13.37)	73.81 (8.59)
+ **CaSpeR**	22.61 (71.01)	29.72 (57.60)	70.97 (11.75)	75.18 (7.93)
X-DER $^{\text{RPC}}_{\text{future}}$	24.80 (44.69)	30.98 (30.12)	74.32 (4.95)	77.70 (3.71)
+ **CaSpeR**	26.24 (41.72)	31.55 (28.71)	75.99 (3.88)	78.71 (2.32)
PODNet	16.82 (52.32)	20.81 (46.50)	60.60 (14.00)	66.15 (10.71)
+ **CaSpeR**	18.09 (50.33)	22.45 (46.08)	64.84 (10.01)	70.85 (7.99)

Tab. 7.2: FAA (FAAF) on S-*mini*Img for RBMs with and w/o CaSpeR.

successfully counteracts the learning bias presented in Sec. 5.1.3.

While still improving over the baselines, we see a reduced FAA improvement on S-*mini*Img. The mixed FAAF results in Class-IL might suggest that our approach is not particularly beneficial when it comes to comparing classes learnt at different tasks. We suspect that this might be a by-product of our approximated batch operation, which only considers a few classes at any given training step and therefore struggles when dealing with the increased number of tasks in this dataset. Even so, the Task-IL values for FAAF are favourably reduced, meaning that CaSpeR lets the model learn individual tasks more accurately so that it aptly recovers its predictive capability when cued with the correct task.

As a final note, PODNet appears to be an outlier; with lower FAA and higher FAAF w.r.t. the other evaluated approaches, it exhibits a marked tendency to overfit current training data. Nevertheless, CaSpeR is still capable of impacting its training positively, delivering a stabilising effect that is especially relevant when the memory buffer is large. This suggests that the additional regularisation facilitates the model's convergence, which aligns with the observations we make in Sec. 9.5.3, where we will exploit CaSpeR with limited supervision.

7.4 Analysis

In this section, we briefly present two additional experiments aimed at showing the geometric properties conferred to the model by CaSpeR.

k-NN Clsf (Class-IL)	w/o CaSpeR 5-NN	w/o CaSpeR 11-NN	w/ CaSpeR 5-NN	w/ CaSpeR 11-NN
ER-ACE	43.73	44.41	46.75+3.02	47.29+2.88
iCaRL	34.86	37.78	36.00+1.14	38.33+0.55
DER++	44.21	44.24	45.75+1.54	46.00+1.76
X-DER $^{\text{RPC}}_{\text{future}}$	43.44	44.62	49.47+6.03	49.49+4.87
PODNet	21.11	22.60	27.88+6.77	28.94+6.34

Tab. 7.3: Class-IL FAA results (S-CIFl00, $|\mathcal{M}| = 2000$) of k-NN classifiers trained on top of the latent representations of $\mathcal{M}$ data.

7.4.1 k-NN classification

As a first concern, we desire to verify whether CaSpeR succeeds in separating the latent embeddings for examples of different classes. The results presented in Fig. 7.1 already verified this assumption by analysing the Label-Signal Variation (σ) on $\mathcal{G}$; here, we further evaluate this aspect by training k-NN-classifiers [183] on top of the latent representations g produced by the methods of Sec. 7.3 and evaluating their FAA.

In Tab. 7.3, we report the results for 5-NN and 11-NN classifiers which use the latent-space projections of the final buffer $\mathcal{M}$ as a support set. We observe that the steady beneficial effect shown by CaSpeR in our previous experiments also extends to this distinctive classification approach. This constitutes a confirmation that our proposal is instrumental in disentangling the representations of different classes.

7.4.2 Latent Space Consistency

To provide further insights into the dynamics of the latent space on the evaluated models, we study the emergence of distortions in the LGG. Given a RBM, we are interested in a comparison between $\mathcal{G}_4$ and $\mathcal{G}_9$, the LGGs produced after training on $\mathcal{T}_4$ and $\mathcal{T}_9$ respectively, computed on the test set of tasks $\mathcal{T}_0, ..., \mathcal{T}_4$.

The comparison between $\mathcal{G}_4$ and $\mathcal{G}_9$ can be better understood in terms of the node-to-node bijection $\mathfrak{F} : \mathcal{G}_4 \to \mathcal{G}_9$, which can be represented as a functional map matrix C [125] with elements

$$c_{i,j} \triangleq \langle \phi_i^{\mathcal{G}_4}, \phi_j^{\mathcal{G}_9} \circ \mathfrak{F} \rangle \,, \tag{7.5}$$

where $\phi_i^{\mathcal{G}_4}$ is the i^{th} Laplacian eigenvector of $\mathcal{G}_4$ (similarly for $\mathcal{G}_9$) and $\circ$ denotes the standard function composition. In other words, the matrix C encodes the similarity between the Laplacian eigenspaces of the two graphs. In an ideal scenario where the latent space is subject to no

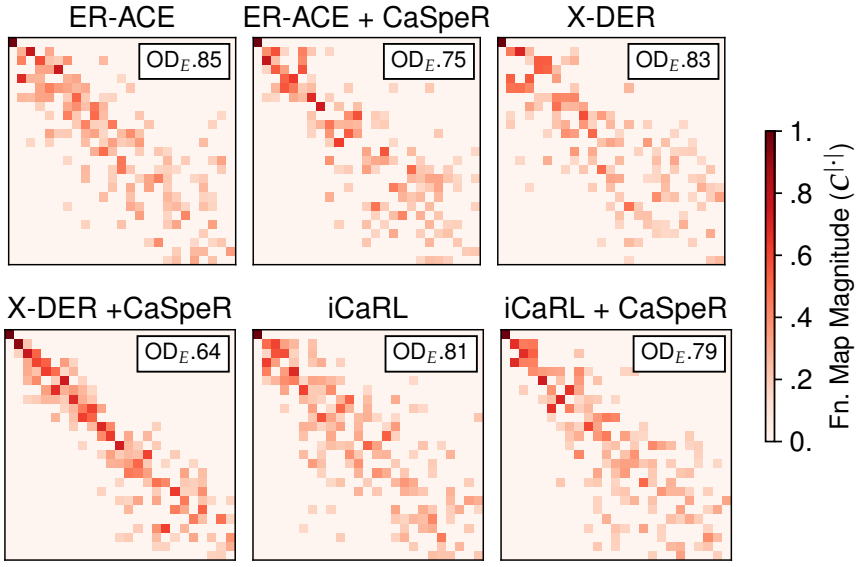


Fig. 7.3: For several RBMs with and without CaSpeR, the functional map magnitude matrices $C^{|\cdot|}$ between the LGGs $\mathcal{G}_4$ and $\mathcal{G}_9$, computed on the test set of $\mathcal{T}_0, ..., \mathcal{T}_4$ after training up to $\mathcal{T}_4$ and $\mathcal{T}_9$ respectively (S-CIF100 - $|\mathcal{M}| = 2000$). The closer $C^{|\cdot|}$ to the diagonal, the less geometric distortion between $\mathcal{G}_4$ and $\mathcal{G}_9$. We report the first 25 rows and columns of $C^{|\cdot|}$, focusing on smooth (low-frequency) matches [125] and apply a $C^{|\cdot|} > 0.15$ threshold to increase clarity.

modification between $\mathcal{T}_4$ and $\mathcal{T}_9$ w.r.t. previously learnt classes, $\mathfrak{F}$ is an *isomorphism* and C is a diagonal matrix [125]. In a practical scenario, $\mathfrak{F}$ is only approximately isomorphic and, the better the approximation, the more C becomes sparse and funnel-shaped.

In Fig. 7.3, we report $C^{|\cdot|} \triangleq \text{abs}(C)$ for ER-ACE, DER++, iCaRL and X-DER $^{\text{RPC}}_{\text{future}}$ on S-CIF100, both with and without CaSpeR. It can be observed that the methods that benefit the most from our proposal (ER-ACE, X-DER $^{\text{RPC}}_{\text{future}}$) display a tighter functional map matrix. This indicates that the partitioning behaviour promoted by CaSpeR leads to reduced interference, as the portion of the LGG that refers to previously learnt classes remains geometrically consistent in later tasks. On the other hand, in line with the considerations made in Sec. 7.3, the improvement is only marginal for iCaRL; its different training regime, which is less discriminative in nature, seemingly induces a limited amount of change on the structure of the latent space.

To quantify the similarity of each $C^{|\cdot|}$ matrix to the identity, we also

report its off-diagonal energy, computed as follows [145]:

$$\mathrm{OD}_E \triangleq \frac{1}{||C||_{\mathrm{F}}^2} \sum_i \sum_{j \neq i} c_{i,j}^2, \tag{7.6}$$

where $||\cdot||_{\mathrm{F}}$ indicates the Frobenius norm. CaSpeR produces a clear decrease in OD_E, signifying an increase in the diagonality of the functional matrices.

7.5 Conclusions

This chapter introduced CaSpeR, a regularising constraint on RBMs' latent space aimed at encouraging a clustering behaviour on $\mathcal{M}$. The proposed approach exploits spectral geometry to allow for an easy manipulation of the model's latent space and produces a quantifiable disentanglement of the latent projections of points belonging to distinct classes.

As CaSpeR does not rely on the availability of annotations for each example, it can easily be applied to scenarios where only limited supervision is available. The main approach proposed in Chap. 9 for dealing with a reduced-annotation CL scenario encourages coherent class representations in a similar spirit to CaSpeR (albeit without resorting to a geometrical formulation of its learning objective). Furthermore, the same chapter will introduce an additional experiment on CaSpeR, showcasing how its objective provides better accuracy and easier convergence when data annotations are scarce.

The exploitation of geometric constraints on continual learners appears to be a high-potential research direction, of which this chapter only represents an initial exploration. Our preliminary investigations suggest that the latent-space entanglement effects mentioned here are particularly severe in unsupervised continual learning scenarios [106, 46] due to the weak training signal caused by the lack of annotations. These settings should therefore provide a natural testbed for the application of spectral and (more broadly) geometric regularisers aiming at endowing the model with desirable properties.

Part III

Beyond Basic Continual Learning Settings

Chapter 8

General Continual Learning

8.1 Motivation

When starting the study of CL, one easily comes to realise that the majority of methods proposed in the classical CL literature are hardly suited for real-world applications. In line with the many works that recently proposed leaving the *academic* CL scenarios behind in favour of more realistic experimental settings, this Chapter also proposes a more challenging experimental benchmark.

Our proposal was inspired by the General Continual Learning (GCL) guidelines featured in Sec. 7 of [35]. The authors of this work outlined a series of desiderata of an ideal CL evaluation scheme, general enough to allow for the modelling of input data-streams akin to the ones found in a practical scenario, possibly characterised by sudden or gradual changes. The original list of desiderata consists of the following 10 key points:

1. **Constant Memory**: have a bounded memory footprint;
2. **No Task Boundaries**: do not rely on boundaries between tasks;
3. **Online Learning**: do not perform offline training on large batches of data;
4. **Forward Transfer**: facilitate the learning of new tasks by means of previously acquired knowledge;
5. **Backward Transfer**: improve previous knowledge while learning new tasks;
6. **Problem Agnostic**: do not limit the problem to one setting (*e.g.*, classification);
7. **Adaptive**: allow learning from unlabelled data;

CL methods *vs* GCL requirements	PNN	PackNet	HAT	ER	MER	GSS	GEM	A-GEM	HAL	iCaRL	FDR	LwF	SI	oEWC	**DER**	**DER++**
Constant memory	–	–	–	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
No task boundaries	–	–	–	✓	✓	✓	–	†	–	–	–	–	–	–	✓	✓
Online learning	✓	–	✓	✓	✓	✓	✓	✓	–	–	✓	✓	✓	–	✓	✓
No test time oracle	–	–	–	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	✓	✓

Tab. 8.1: CL approaches and their compatibility with the major GCL requirements [35]. † indicates compatibility provided that the model adopts *reservoir* sampling instead of the original *ring* strategy.

8. **No Test Time Oracle**: do not require task identifiers at inference time;
9. **Task Revisiting**: allow for the enhancement of acquired knowledge by revisiting past tasks;
10. **Graceful Forgetting**: allow for selective forgetting of trivial information to retain a balance between stability and plasticity.

Among these requirements, points 4, 5 and 10 provide methodological suggestions that are generally measured *post-hoc* via the metrics presented in Sec. 2.4; points 6, 7 and 9 concern recommendations for designing CL experiments; finally, points 1, 2, 3 and 8 are the only ones that directly pertain to the design of CL methods.

We investigate the compatibility of existing CL approaches with the last group of requirements and, as reported in Table 8.1, remarkably find that very few methods in literature meet all the criteria to operate in a GCL scenario. Indeed, traditional *distillation-based methods* depend on the task boundary to backup the teacher model (or store teacher responses, which is not an online operation); *architectural methods* allow for some degree of model growth, incurring in non-constant memory requirements; *regularisation methods* perform cumbersome parameter-importance estimation, possibly at a non-constant memory requirement. This only leaves RBMs, barring out those that strictly require task identities while training (GEM) or perform cumbersome operations at task boundaries (HAL).

Having established which approaches are suitable for GCL, we set out to compare them experimentally on a novel benchmark inspired by

Fig. 8.1: Example batches of the MNIST-360 stream.

the desiderata of [35]. For this reason, we introduce MNIST-360, a novel MNIST-based benchmark featuring task revisiting and entailing both gradual and sudden changes in the input distribution.

8.2 MNIST-360

MNIST-360 is a novel experimental protocol targeting the GCL setting. It models a stream of data presenting batches of two consecutive MNIST [88] digits at a time (*i.e.*, $\{0,1\}$, $\{1,2\}$, $\{2,3\}$, etc.), as depicted in Fig. 8.1. After a fixed number of steps, we switch the lesser of the two digits with the following one, introducing a distribution shift similar to those encountered at task boundaries in Class-IL or Task-IL. Additionally, each example shown on the stream is rotated by an increasing angle, which produces a gentle but constant distribution shift at every input step.

As it is impossible to distinguish 6s and 9s upon rotation, we exclude 9s from MNIST-360; this means that the stream can show nine possible pairs of classes. Each pair is shown more than once, allowing the model to leverage positive transfer upon revisiting the same classification problem. To address the implicit assumption of real-world CL that a specific input data-point would only be observed once [105, 141, 26], we enforce the following guarantees: *i)* each example is only shown once during the overall training; *ii)* digits of the same class are never observed under the same rotation.

In light of the described characteristics of MNIST-360, it should be noted that the benchmark does not present distinguished tasks, but rather a unified input stream without interruptions. Furthermore, the intertwining of sharp (change in class) and smooth (rotation) distribution shifts makes their detection hard for algorithms that might try to identify them from data.

In the following, we present the technical details for the construction of the training and test set of MNIST-360.

8.2.1 Training Set

For Training purposes, we build batches using exemplars that belong to two consequent classes at a time, meaning that 9 pairs of classes are possibly encountered: $(0,1)$, $(1,2)$, $(2,3)$, $(3,4)$, $(4,5)$, $(5,6)$, $(6,7)$, $(7,8)$, and $(8,0)$. Each pair is shown in this order in R rounds ($R = 3$ in our experiments) at changing rotations. This means that MNIST-360 consists of $9R$ pseudo-tasks, whose boundaries are not signalled to the tested method. We indicate them with $\Psi_r^{(d_1,d_2)}$ where $r \in \{1,\dots,R\}$ is the round number and d_1, d_2 are digits forming one of the pairs listed above.

As every MNIST digit d appears in $2R$ pseudo-tasks, we randomly split its example images evenly in 6 groups G_i^d where $i \in \{1,\dots,2R\}$. The set of exemplars shown in $\Psi_r^{(d_1,d_2)}$ is given as $G_{[r/2]}^{d_1} \cup G_{[r/2]+1}^{d_2}$, where $[r/2]$ is an integer division.

At the beginning of $\Psi_r^{(d_1,d_2)}$, we initialise two counters C_{d_1} and C_{d_2} to keep track of how many exemplars of d_1 and d_2 are shown respectively. Given batch size B ($B = 16$ in our experiments), each batch is made up of N_{d_1} samples from $G_{[r/2]}^{d_1}$ and N_{d_2} samples from $G_{[r/2]+1}^{d_2}$, where:

$$N_{d_1} = min\left(\frac{|G_{[r/2]}^{d_1}| - C_{d_1}}{|G_{[r/2]}^{d_1}| - C_{d_1} + |G_{[r/2]+1}^{d_2}| - C_{d_2}} \cdot B, |G_{[r/2]}^{d_1}| - C_{d_1}\right), \tag{8.1}$$

$$N_{d_2} = min\left(B - N_{d_1}, |G_{[r/2]+1}^{d_2}| - C_{d_2}\right). \tag{8.2}$$

This allows us to produce balanced batches, in which the proportion of exemplars of d_1 and d_2 is kept fixed. Pseudo-task $\Psi_r^{(d_1,d_2)}$ ends when the entirety of $G_{[r/2]}^{d_1} \cup G_{[r/2]+1}^{d_2}$ has been shown, which does not necessarily happen after a fixed number of batches.

Each digit d is also associated with a counter C_d^r that is never reset during training and is increased every time an exemplar of d is shown to the evaluated method. Before its showing, every exemplar is rotated by

$$\frac{2\pi}{|d|} C_d^r + O_d \tag{8.3}$$

where $|d|$ is the number of total examples of digit d in the training set and O_d is a digit-specific angular offset, whose value for the i^{th} digit is given by $O_i = (i-1)\,\pi/2R$ ($O_0 = -\pi/2R$, $O_1 = 0$, $O_2 = \pi/2R$, etc.). By so doing, all digit's exemplars are shown with an increasing rotation spanning a 2π angle throughout the entire procedure. Rotation changes within each pseudo-task, resulting into a gradually changing distribution. Fig. 8.1 the first batch of the initial 24 pseudo-tasks with $B = 9$.

FA	MNIST-360		
JT	82.98		
FT	19.09		
$\|\mathcal{M}\|$	200	500	1000
ER	49.27	65.04	75.18
MER	48.58	62.21	70.91
A-GEM-R	28.34	28.13	29.21
GSS	43.92	54.45	63.84
DER (ours)	**55.22**	69.11	75.97
DER++ (ours)	54.16	**69.62**	**76.03**

Tab. 8.2: Accuracy of compatible RBMs on the test set of MNIST-360.

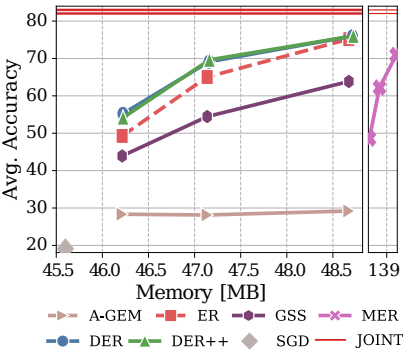


Fig. 8.2: Results on MNIST-360 also characterised w.r.t. the memory footprint of the evaluated approaches.

8.2.2 Test Set

As no task boundaries are provided, evaluation on MNIST-360 can only be carried out after the training is complete. For test purposes, digits are still shown with an increasing rotation as per Eq. 8.3, with $|d|$ referring to the test-set exemplar count and no offset applied ($O_d = 0$).

The order with which digits are shown is irrelevant, hence no specific batching strategy is needed and we simply show one digit at a time.

8.3 Experiments

In this section, we present an experimental comparison on MNIST-360 of the RBMs that are compatible with GCL (*i.e.*, ER, MER, GSS, DER and DER++), with the addition of a variant of A-GEM equipped with a *reservoir* memory buffer (A-GEM-R). These methods are evaluated at $|\mathcal{M}| \in \{200, 500, 1000\}$ through 10 independent experimental runs. We employ the same backbone design used for experiments on MNIST-based benchmarks in Tab. 2.1: a 2-layer MLP with hidden size 100. Results are provided in terms of simple Final Accuracy (FA), since there are no tasks among which to compute an average. Fig. 8.2 further characterises these results in terms of memory occupancy; this shows that all evaluated methods entail a negligible overhead w.r.t. the FT baseline, with the sole exception of MER. Due to its meta-update scheme, this last approach needs to retain three copies of the model's parameters, which makes it largely impractical for application on larger-scale benchmarks.

8.4 Conclusions

This brief chapter presented a novel experimental setting, originally designed to showcase DER's ability to operate in a realistic setting, in line with the guidelines expressed in [35]. While it could be argued that other *novel* scenarios presented in Sec. 2.2.2 enjoy wider popularity, our proposed benchmark is designed from the ground up to match the key requirements of GCL and ensure that incompatible methods simply cannot run the experiment.

While several works have proposed evaluations on our MNIST-360 after its first publication [9, 187, 17, 94, 176, 66], no additional benchmarks designed specifically for GCL have been proposed. While the performance achieved by these models has not yet saturated for low $|\mathcal{M}|$ w.r.t. to JT, there is reason to believe that the CL community would particularly benefit from the design of a similar GCL experiment entailing a more complex base dataset than MNIST.

Chapter 9

Continual Learning under Limited Supervision

9.1 Motivation

Both *academic* and novel CL scenarios presented in Chap. 2 make the assumption that all incoming data is labelled. In some settings, this condition does not represent an issue and can be easily met. This may be the case when ground-truth annotations can be directly and automatically collected (*e.g.*, a robot that explores the environment and learns to avoid collisions by receiving direct feedback from it [5]).

However, if the labelling stage involves human intervention (as it is the case for a number of computer vision tasks including classification, object detection [207], etc.), the assumption of full supervision clashes with the pursuit of lifelong learning. Indeed, if the adaptability of the learner to incoming tasks were bottlenecked by the speed of the human annotator, the trivial solution of re-training from scratch would become a viable alternative to continually updating the model. For this reason, in this chapter we propose a scenario called **Continual Semi-Supervised Learning (CSSL)**, which accounts for annotations being made available to the learner at a reduced rate. Specifically, we assume that only **one out of k** examples is presented with its ground-truth label.

To address this setting, one could simply limit the adjustment of the prediction model to the fraction of examples that can be labelled in real-time; we empirically review the performance of SOTA CL models at varying label-per-example rates and verify that doing so results in an expected degradation in terms of performance. Luckily, this can be compensated by leveraging techniques from *semi-supervised learning* [124, 166]: many of the research efforts in this field can be beneficially combined with CL models to allow learning even from unlabelled observations.

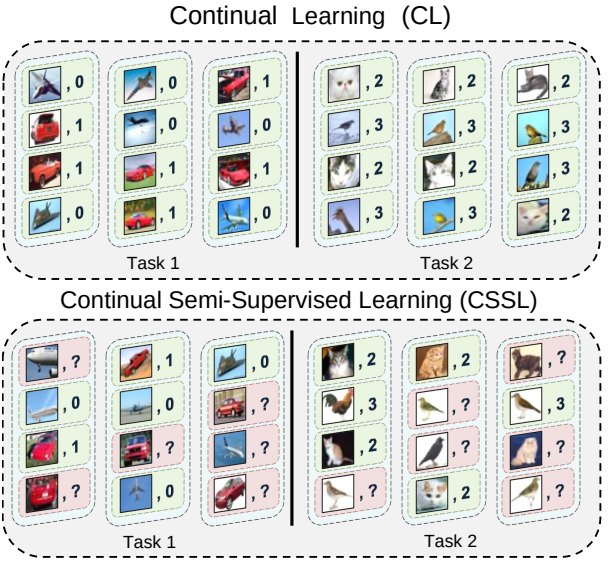


Fig. 9.1: Overview of the CSSL setting. Input batches include both labelled (*green*) and unlabelled (*red*) examples.

Taking one more step, we then also propose a CSSL method capable of filling the gap induced by partial annotations. **Contrastive Continual Interpolation Consistency (CCIC)**, enforces the consistency of augmented and interpolated examples and encourages coherent class representations. We surprisingly find that CCIC is not necessarily upper-bounded by fully supervised learners: 25% labels can be enough to outperform CL methods using all ground truth.

9.2 Continual Semi-Supervised Learning

In CSSL, we propose a variation of the CL problem presented in Sec. 2.1 by distributing the samples coming from $\mathcal{T}_t$ into two sets: $\mathcal{T}_t^s$, containing a limited number of pairs of labelled samples (x^s, y^s), and $\mathcal{T}_t^u$, containing the remaining unsupervised samples x^u. We define this split according to a given proportion $p_s = |\mathcal{T}_t^s| / (|\mathcal{T}_t^s| + |\mathcal{T}_t^u|)$, which remains fixed across all tasks. The objective of Eq. 2.2 must then be optimised without having access to the ground-truth supervision signal for $\mathcal{T}_t^u$. Each input batch of data from the stream consists of a mix of labelled pairs $\mathcal{S} \subset \mathcal{T}_t^s$ and unlabelled items $\mathcal{U} \subset \mathcal{T}_t^u$; since input batches are randomly sampled from the available data, we can equivalently formulate CSSL as providing a ground-truth label for any given stream example with uniform probability $1/k$ (as shown in Fig. 9.1 for $k = 2$).

In line with the novel CL scenarios presented in Sec. 2.2.2, this chapter also aims at providing a more realistic setup: instead of focusing on model limitations, we acknowledge that requiring fully labelled data can hinder the extension of CL algorithms to real-time and in-the-wild applications.

Some recent works that also focus on exploiting unlabelled data in CL methods are **Deep Model Consolidation** [203], which first specialises a dedicated model on each new encountered task, then produces a unified learner by distilling knowledge from both the new specialist and the previous incremental model, and **Semi-Supervised Incremental Learning** [87], which alternates unsupervised feature learning on both input and auxiliary data with supervised classification. We remark that both these settings are significantly different from our proposed **CSSL**: we do not separate the supervised and unsupervised training phases, but rather intertwine both kinds of data in all drawn batches in varying proportions and require that the model learns from both at the same time. Additionally, we do not exploit auxiliary unsupervised external data to supplement the training set; instead, we reduce the original supervised data to a fraction, thus modelling supervision becoming available on the input stream at a much slower rate.

9.3 CSSL approaches

We are interested in understanding *i)* how existing CL perform under partial lack of supervision and *ii)* how Semi-Supervised Learning approaches can be combined with them to exploit unsupervised data. Question *i)* is investigated experimentally in Sec. 9.4 by evaluating methods that simply *drop* unlabelled examples x^u. Differently, question *ii)* opens up many possible solutions that we address by proposing a simple CSSL baseline first (PseudoER) and then a more complex approach (CCIC). Due to the effectiveness of RBMs shown in previous chapters, we choose to build both proposals on top of the lightweight ER.

9.3.1 Pseudo-Labelling Experience Replay

Inspired by the line of works relying on self-labelling [191, 89], we first introduce **PseudoER**: a simple CSSL baseline allowing ER to profit from the unlabelled examples. When dealing with lack of supervision, the model itself can be used to produce targets (*pseudo-labels*) [191, 89]; given an example x^u without annotation, PseudoER produces a *pseudo-label* $\tilde{y}^u$ by considering its own prediction on x^u. Formally,

$$\tilde{y}^u = \underset{c \in \mathcal{Y}_c}{\operatorname{argmax}} f_\theta(x^u)_c, \tag{9.1}$$

where $\mathcal{Y}_c$ is the set of classes of the current task.

Alg. 9.1: Contrastive Continual Interpolation Consistency

Input: Input batch $\mathcal{X} = \mathcal{X}_s \cup \mathcal{Y}_s \cup \mathcal{X}_u$ (superv. samples $x_s \in \mathcal{X}_s$, labels $y_s \in \mathcal{Y}_s$, unsup. items $x_u \in \mathcal{X}_u$), memory buffer $\mathcal{M}$, scalars λ, μ, τ, α, β, weights θ.
$(\mathcal{X}_M, \mathcal{Y}_M) \leftarrow \text{sampleBatch}(\mathcal{M})$
$\mathcal{S} \leftarrow (\text{augment}(x_s), \forall x_s \in [\mathcal{X}_s, \mathcal{X}_M])$
$\mathcal{U}, \mathcal{P} \leftarrow [\,], [\,]$
for x_u **in** $\mathcal{X}_u$ **do**
  for $k = 1$ **to** K **do**
    $\hat{x}_{u,k} \leftarrow \text{augment}(x_u)$
    $\mathcal{U} \leftarrow [\mathcal{U}, \hat{x}_{u,k}]$
  end for
  $z_u \leftarrow \sum_k h_\theta(\hat{x}_{u,k})$
  $\tilde{z}_u \leftarrow z_u^{1/\tau} / \sum_j (z_u)_j^{1/\tau}$
  $\mathcal{P} \leftarrow [\mathcal{P}, \text{repeat}(\tilde{z}_u, K)]$
end for
$\mathcal{W} \leftarrow \text{shuffle}([\mathcal{S}, \mathcal{U}])$
$\mathcal{S}', \mathcal{U}' \leftarrow \text{mixUp}(\mathcal{S}, \mathcal{W}^{<|\mathcal{S}|}), \text{mixUp}(\mathcal{U}, \mathcal{W}^{\geqslant|\mathcal{S}|})$
$\mathcal{S}^*, \mathcal{U}^* \leftarrow \text{zip}(\mathcal{S}', [\mathcal{Y}_s, \mathcal{Y}_M]), \text{zip}(\mathcal{U}', \mathcal{P})$
$\mathcal{L}_{\text{S}} \leftarrow \sum_{x,y \in \mathcal{S}^*} \text{CE}\big(f_\theta(x), y\big)$
$\mathcal{L}_{\text{U}} \leftarrow \sum_{u,q \in \mathcal{U}^*} \|q - f_\theta(u)\|_2^2$
$\mathcal{H}_S \leftarrow \{(x, \text{hrdstPositive}(x, \mathcal{S}), \text{hrdstNegative}(x, \mathcal{S})); \forall x \in \mathcal{S}\}$
$\mathcal{L}_{\text{SM}} \leftarrow \sum_{x, x_\text{N}, x_\text{P} \in \mathcal{H}_S} \text{ReLU}(\alpha - \|h_\theta(x) - h_\theta(x_\text{N})\|_2^2 + \|h_\theta(x) - h_\theta(x_\text{P})\|_2^2)$
$\mathcal{H}_U \leftarrow \{(x, \text{hrdst}(x, \mathcal{M}_{<\mathcal{T}_\rfloor})), \forall x \in \mathcal{U}\}$
$\mathcal{L}_{\text{UM}} \leftarrow \sum_{x, x_\text{N} \in \mathcal{H}_S} \text{ReLU}(\beta - \|h_\theta(x) - h_\theta(x_\text{N})\|_2^2)$
$\mathcal{L}_{\text{CCIC}} \leftarrow \mathcal{L}_\text{S} + \lambda \mathcal{L}_\text{U} + \mathcal{L}_\text{SM} + \mu \mathcal{L}_\text{UM}$

Unfortunately, *pseudo-labels* tend to become unstable when only a few annotations are available, possibly resulting in overfitting of the limited supervised data available [123]. To mitigate this effect, we apply a threshold η to disregard x^us with low-confidence outputs. Specifically, we estimate the confidence as the difference between the two highest values pre-softmax model responses $h_\theta(x^u)_c$[1]. After this filtering step, a pair $(x^u, \tilde{y}^u)$ is considered on par with any supervised pair (x^s, y^s) and is therefore inserted into $\mathcal{M}$ for later replay.

9.3.2 Contrastive Continual Interpolation Consistency

If PseudoER represents a simple approach for incorporating unlabelled items into the model's learning, we take one more step and propose **Contrastive Continual Interpolation Consistency (CCIC)**: a more comprehensive CSSL approach that is presented in detail in Alg. 9.1. Our proposal

[1]With h_θ indicating pre-softmax model responses of $f_\theta(\cdot)$, as defined in Sec. 3.3.2.

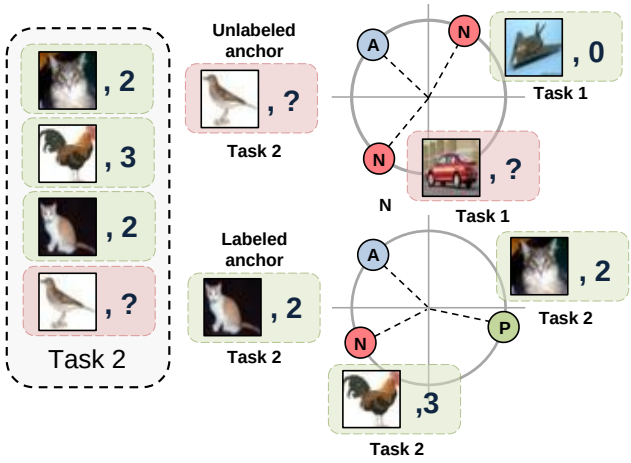


Fig. 9.2: An illustration of the semantic constraints enforced by CCIC: for each anchor (A), the network pushes away representations of different-task and different-class examples (N) and attracts same-class embeddings (P).

combines of ER with **MixMatch** [16], a recently proposed semi-supervised learning approach which does not depend on the model's accuracy on unlabelled examples, making it a more robust approach w.r.t. self-labelling. On top of that, CCIC applies additional consistency objectives to ensure coherent representations for examples belonging to the same class.

Continual MixMatch

Following MixMatch, x^us from the input stream are subjected to different augmentations and their predictions are averaged, sharpened and finally used to promote consistent responses to considerable variations of the data-points. This *consistency regularisation* step [166, 116] operates by combining both labelled and unlabelled samples through the mixUp procedure [201][2]: this produces two final augmented and mixed sets of examples $\mathcal{S}^*$ and $\mathcal{U}^*$ that are then used to compute two loss terms, $\mathcal{L}_\text{S}$ using ground-truth labels of the former set and $\mathcal{L}_\text{U}$ using the soft-labels generated through response-averaging. Please refer to Alg. 9.1 and [16] for a step-by-step breakdown of this procedure.

Inter-Class Consistency

To further push the model to produce semantically coherent predictions, we encourage it to keep the representations of labelled examples belonging

[2]Differently from MixMatch, we found it more effective to only apply mixUp on the input images and not to the corresponding labels.

to the same class possibly close. This is accomplished by introducing a triplet margin loss [11], leveraging positive and negative anchors x_P and x_N chosen as the hardest same-class and different-class exemplars respectively, either within the labelled portion of the current batch $\mathcal{S}$ or the memory buffer $\mathcal{M}$. In formal terms:

$$\mathcal{L}_{\text{SM}} \triangleq \mathop{\mathbb{E}}_{(x,x_N,x_P)\sim\mathcal{S}\cup\mathcal{M}} \Big[\max\big(\alpha - D_\theta(x,x_N) + D_\theta(x,x_P), 0\big)\Big], \tag{9.2}$$

with D indicating the Euclidean distance between the embeddings of the provided samples ($D_\theta(x,x') \triangleq \|h_\theta(x) - h_\theta(x')\|_2^2$) and α a constant margin beyond which no more efforts should be put into enlarging the distance between positive and negative pairs.

Since the objective of Eq. 9.2 must be limited to the few labelled examples at our disposal, we are interested in further refining training by devising a similar contrastive term on unlabelled data. To do so, we recall that we work in the Class-IL setting: tasks are disjoint (*i.e.*, examples from different tasks necessarily belong to different classes) and the boundaries between tasks are provided. We can therefore keep track of the original task of in-memory exemplars thus use stored examples from previous tasks $\mathcal{M}_{<\mathcal{T}_c}$ as negative anchors to current-task unlabelled exemplars $\mathcal{U}$:

$$\mathcal{L}_{\text{UM}} \triangleq \mathop{\mathbb{E}}_{\substack{x\sim\mathcal{U}\\ x_N\sim\mathcal{M}_{<\mathcal{T}_c}}} \Big[\max\big(\beta - D_\theta(x,x_N), 0\big)\Big], \tag{9.3}$$

with β a separate constant margin analogous to α in Eq. 9.2. The combined effect of $\mathcal{L}_{\text{SM}}$ and $\mathcal{L}_{\text{UM}}$ is illustrated in Fig. 9.2.

Overall objective

In summary, the objective of CCIC combines the regularisation of MixMatch with the additional terms given by Eq. 9.2 and 9.3. The overall optimisation target is formalised as follows:

$$\hat{\mathcal{L}}_{\text{CL}} \triangleq \mathcal{L}_{\text{S}} + \lambda\mathcal{L}_{\text{U}} + \mathcal{L}_{\text{SM}} + \mu\mathcal{L}_{\text{UM}}, \tag{9.4}$$

where λ and μ are hyper-parameters setting the importance of the unsupervised examples.

Inference Scheme

To maximally exploit the introduced feature-space constraints of Eq. 9.2 and 9.3, we further propose an alteration of the basic inference scheme of ER. Similarly to [138], we decouple classification from feature extraction by employing the k-Nearest Neighbours classifier trained on top of the memory buffer for as final predictor. This is in harmony with the rest of the model and further mitigates the bias problem discussed in Sec. 5.1.3.

FAA	**S-SVHN** (JT: 86.2)				**S-CIF10** (JT: 92.1)				**S-CIF100** (JT: 67.7)			
Labels %	*0.8%*	*5%*	*25%*	*100%*	*0.8%*	*5%*	*25%*	*100%*	*0.8%*	*5%*	*25%*	*100%*
FT	9.9	9.9	17.5	17.8	13.6	18.2	19.2	19.6	1.8	5.0	7.8	8.6
LwF	9.9	9.9	14.8	16.9	13.1	17.7	19.4	19.6	1.6	4.5	8.0	8.4
oEWC	9.9	9.9	14.7	17.9	13.7	17.6	19.1	19.6	1.4	4.7	7.8	7.8
SI	9.9	10.2	17.1	18.2	12.4	15.9	19.2	19.5	1.3	3.4	7.5	8.1
$\lvert\mathcal{M}\rvert = 500$												
ER	32.5	56.0	59.6	66.5	36.3	51.9	60.9	62.2	8.2	13.7	17.1	21.3
iCaRL	8.9	10.0	19.9	23.1	24.7	35.8	51.4	61.0	3.6	11.3	**27.6**	**37.8**
DER	11.9	54.6	56.9	**70.8**	29.1	35.3	50.0	**67.1**	1.7	5.1	13.0	28.8
GDumb	34.6	41.8	59.2	59.9	39.5	40.9	44.8	47.9	8.6	9.9	10.1	10.9
PseudoER	23.2	48.9	63.5	-	37.8	44.9	56.3	-	5.1	14.3	18.4	-
CCIC	**55.3**	**70.1**	**75.9**	-	**54.0**	**63.3**	**63.9**	-	**11.5**	**19.5**	20.3	-
$\lvert\mathcal{M}\rvert = 5120$												
ER	44.4	69.9	77.6	**80.5**	37.4	64.1	79.7	83.3	9.6	22.8	37.9	**49.0**
iCaRL	9.3	11.5	19.5	23.9	20.7	35.5	56.3	61.9	4.3	12.2	30.9	41.1
DER	23.1	67.8	74.7	75.3	32.9	47.6	73.9	**84.5**	1.6	4.7	11.9	38.6
GDumb	46.5	74.4	74.6	78.3	40.8	71.2	81.4	82.5	9.6	23.3	33.2	42.9
PseudoER	45.8	74.6	77.8	-	**62.2**	72.9	80.4	-	8.2	25.1	40.0	-
CCIC	**59.3**	**81.0**	**83.9**	-	55.2	**74.3**	**84.7**	-	**12.0**	**29.5**	**44.3**	-

Tab. 9.1: Class-IL FAA of CL methods and our proposals in CSSL.

9.4 Experiments

In this section, we provide an evaluation encompassing several CL methods and our CSSL proposals on CSSL benchmarks. The latter are constructed by taking standard Class-IL benchmarks from Sec. 2.3 and randomly keeping only a percentage of the ground-truth labels, while discarding the rest. We employ S-SVHN, S-CIF10 and S-CIF100[3] and vary the fraction of labelled data shown to the model to test different degrees of supervision (0.8%, 5%, 25% and 100%, *i.e.*, 400, 2500, 25000, and 50000 samples on S-CIF10/S-CIF100); results are expressed as FAA and averaged over 5 independent runs.

For each benchmark, we provide an upper bound given by JT without discarding any ground-truth annotation and a lower bound given by FT. To test existing CL methods on our settings, we consider LwF, oEWC, SI, ER, iCaRL, DER, GDumb and simply discard the unlabelled examples in each input batch.

[3]W.r.t. to the details presented in Sec. 2.3, we only train for 30 epochs on S-CIF100 and use no learning rate decay; all models are optimised with SGD with the sole exception of CCIC, which uses Adam.

9.4.1 CL Models

The results presented in Tab. 9.1 confirm that CSSL constitutes a challenging scenario, whose difficulty unsurprisingly increases when fewer labels are provided to the learner. Regularisation methods – generally regarded as weak on Class-IL even with full supervision [45, 6] – dramatically underperform across all datasets. As these methods rarely outperform the FT lower bound, they prove ineffective outside of Task-IL and Domain-IL in the low-label regime.

RBMs overall show an expected decrease in performance as supervision diminishes. This is especially severe for DER and iCaRL, as their accuracy drops on average by more than 70% between 100% and 0.8% labels. As the model underfits the task when less supervision is provided, it produces less reliable targets that cannot be successfully used by these knowledge distillation-based methods. In contrast, ER is able to replay information successfully as it exploits hard targets; thus, it learns effectively even after initially underfitting the task. Indeed, its accuracy with 5% labels and $|\mathcal{M}| = 5120$ is always higher than its fully supervised accuracy with a smaller buffer, indicating that ER is able to overcome the lack of labels when paired with an appropriate buffer.

We attribute the failure of iCaRL on S-SVHN to the low complexity of the backbone network: a shallow backbone produces a latent space that is less suitable for its nearest-mean-of-exemplars classifier. Conversely, this method proves quite effective even with a reduced memory buffer on S-CIF100. In this benchmark, the *herding sampling* of iCaRL ensures that all classes are fairly represented.

Finally, GDumb does not suffer from lower supervision as long as its buffer can be filled completely: its operation is not disrupted by unlabelled examples on the stream, which is ignored entirely. While this approach outperforms other methods when few labels are available, CCIC surpasses it consistently. This suggests that the stream offers potential for further learning and should not be dismissed.

9.4.2 CSSL Models

Our PseudoER baseline performs notably well on S-CIF10, maintaining high accuracy as the amount of supervision decreases. However, while S-CIF10 is a nontrivial benchmark, it only features two classes for each task, which makes it easy for pseudo-labelling to produce reasonable responses (a random guess results in 50% accuracy). Conversely, PseudoER struggles to produce valid targets and exhibits a swift performance drop on S-CIF100 as the availability of labelled data decreases. Similarly, we find the application of pseudo-labelling beneficial for S-SVHN only as the space reserved for the buffer increases, demonstrating the mixed reliability of this approach in the online setting.

FAA ($\lvert\mathcal{M}\rvert = 5120$)	**Labels %** *5%*	*25%*
Across-Task (Eq. 9.3)	29.5	44.3
Within-Task	29.3	44.0
Task-Agnostic	29.1	43.9

Tab. 9.2: Class-IL FAA of distinct unsupervised mining techniques for CCIC on S-CIF100.

Labels %	*0.8%*	*5%*	*25%*
ER+EMA$_{500}$	21.37	26.31	43.25
CCIC$_{500}$	53.96	63.29	63.86
ER+EMA$_{5120}$	25.85	40.77	64.75
CCIC$_{5120}$	55.19	74.34	84.74

Tab. 9.3: Comparison of CCIC with a CSSL approach enforcing temporal consistency (Class-IL FAA on S-CIF10)

On the contrary, the compelling performance of CCIC points to a successful blending of supervised information and semi-supervised regularisation. While ER encounters an average performance drop of 47%, going from 25% to 0.8% labels on S-CIF10, CCIC only loses 26% on average. Surprisingly, we observe that – for the majority of evaluated benchmarks – 25% supervision is enough to approach the results of fully supervised methods, even outperforming benchmarked CL models in some circumstances (S-CIF10 with $\lvert\mathcal{M}\rvert = 5120$, S-SVHN with $\lvert\mathcal{M}\rvert \in \{500, 5120\}$).

This suggests that, when learning from a stream of data, striving to provide full supervision is not as essential as it might be expected: differently from the offline scenario, a larger number of labels might not produce a proportionate profit due to *catastrophic forgetting*. In this respect, our experiments suggest that pairing few labelled examples with semi-supervised techniques represents a more efficient paradigm to achieve satisfactory performance.

9.5 Analysis

In this section, we briefly review the design choices for CCIC and consider alternative approaches for addressing the CSSL setting.

9.5.1 Unsupervised Mining in CCIC

In its unsupervised mining loss term $\mathcal{L}_{\text{UM}}$, CCIC takes examples of previous tasks in $\mathcal{M}$ as negatives (*Across-Task Mining*) and requires their representations to be pushed away from current data. In Tab. 9.2, we compare this design choice with two alternative strategies: *i) Within-Task Mining*, where we let the model choose the negatives from the current task only; and *ii) Task-Agnostic Mining*, where the model can freely pick a negative example from either the memory or the current batch without any task-specific prior. As can be observed, Task-Agnostic Mining and Within-Task Mining lead to a small but consistent decrease in performance, while $\mathcal{L}_{\text{UM}}$ proves to be the most rewarding strategy.

FAA	w/o CaSpeR		w/ CaSpeR	
Labels %	0.8%	5%	0.8%	5%
ER-ACE	8.46	11.87	8.55+0.09	14.16+2.29
PseudoER-ACE	2.31	16.35	9.69+7.38	17.42+1.07
CCIC	11.53	19.52	12.22+0.72	20.32+0.82

Tab. 9.4: Class-IL FAA on S-CIF100 for several methods with and without CaSpeR ($|\mathcal{M}| = 2000$).

9.5.2 Model-Driven Consistency

As an alternative to combining a consistency regularisation objective with ER, we propose an additional *temporal consistency* baseline which requires the activations of the model to match a slower Exponential Moving Average (EMA) checkpoint [166]. Results in Tab. 9.5 show, however, that such approach performs significantly worse than CCIC and even yields worse results w.r.t. ER. This suggests that exponential moving average approaches do not necessarily scale to CL scenarios.

9.5.3 Spectral Regularisation and CSSL

In Sec. 7.3, we highlighted that CaSpeR – our geometrically-based regulariser for RBMs – should be expected to operate well in a low-data regime and facilitate the convergence of underperforming baselines. For this reason, we here conduct an experiment testing its applicability to CSSL.

In a supervised CL setting, we apply CaSpeR to buffer data-points to encourage the separation of all previously encountered classes in the latent space. However, we remark that our approach does not have strict supervision requirements, as it does not need labels to be attached to each node of the LGG, but rather just the total number of classes $c_{\mathcal{M}}$ that must be clustered (ref. Eq. 7.3).

We can therefore adopt CaSpeR in a limited-supervision setting, leveraging its reduced need for supervision to introduce as an additional learning objective computed on all input stream exemplars – both labelled and unlabelled. Specifically, we compute the LGG for current batch exemplars and minimise its first k eigenvalues, with k equal to the number of classes in a given task.

In Tab. 9.4, we report the results of an experiment on S-CIF100 in the CSSL setting with only 0.8% or 5% annotated labels. We compare ER-ACE, an improved version of PseudoER that also adopts Asymmetric Cross-Entropy (PseudoER-ACE) and CCIC with and without CaSpeR on input stream batches. We observe that introducing CaSpeR leads to an overall improvement of all tested models and – most significantly –

counteracts the failure case incurred by PseudoER-ACE applied on top of 0.8% annotated data. In this case, PseudoER-ACE backfires as the provided supervision does not suffice for the learner to produce reliable responses, which compromises the quality of the self-labelling procedure. In this regard, CaSpeR effectively manages to limit the impact of the noisy labels produced by *pseudo-labelling* and delivers a very significant performance increase which reverts this failure case.

9.6 Conclusions

This chapter presented CSSL, a CL setting which questions the strong assumption of the availability of full supervision and – in so doing – facilitates the study of realistic deployment scenarios, in which a human annotator might constitute a bottleneck for an incremental learner. CSSL is easily obtained by dropping a portion of the labels in an ordinary Class-IL benchmark; our experiments show that strong fully supervised CL methods cannot always generalise well to this scenario (*e.g.*, DER and iCaRL) and that a simple *self-labelling* baseline might also encounter failure cases due to the changing input data distribution.

Motivated by these considerations, we proposed CCIC, a first approach designed specifically for CSSL which combines MixMatch and ER and further introduces additional semantic-constraint loss terms. By proving its effectiveness on the newly proposed setting, CCIC provides a stepping-stone for researchers and practitioners interested in bridging the gap between theoretical CL and applied systems.

Since our proposal, the investigation of semi-supervised CL proved to be a niche but steadily researched topic [160, 112, 72]. A parallel emerging research trend is represented by fully unsupervised CL scenarios [106, 46], although it should be noted that the complete absence of supervision dramatically complicates the challenge of training a continual learner; our preliminary experiments in this field appear inconsistent: enlarging $|\mathcal{M}|$ in RBMs does not increase their performance and SOTA methods seem to entirely fail to produce any sensible organisation of the latent space, raising the question if they are learning at all. In light of these considerations, we feel that CSSL can better serve as a launchpad for real-world implementations of CL problems w.r.t. fully unsupervised CL.

Chapter 10

Pre-Training in Continual Learning Classification

10.1 Motivation

In Chap. 9, we challenged the problematic assumption made by typical CL settings that all incoming data is labelled by exploring strategies that allow for learning even when a fraction of the original supervision is available. A widely adopted alternative solution for learning with limited supervision is given by transferring and re-using knowledge across different data domains as typically done in Transfer Learning [126]. In this respect, the simplest approach is given by pre-training the model on a labelled *source* dataset and then finetuning it on the *target* task [195, 140, 55]. Alternatively, more sophisticated domain adaptation algorithms have been proposed [31, 104, 103] mainly based on the concept of *feature alignment* (*i.e.*, reducing the discrepancy between the feature distributions of the target and source domains). Unfortunately, these approaches assume the availability of the source dataset while training, which clashes with the typical constraints imposed in CL scenarios.

Mehta et al. [110] proposed a first analysis the entanglement between CL and pre-training, highlighting that the latter leads the optimisation towards wider minima of the loss landscape – a property strictly linked to a reduced tendency in incurring forgetting, as discussed in previous chapters. Here, we further the investigation and show a notable effect: the pre-training task is itself catastrophically forgotten as the model veers towards the newly introduced stream of data. This is not detrimental if all target classes are available at once (*i.e.*, JT): as their exemplars can be accessed simultaneously, the learner can discover a joint feature alignment that works well for all of them while leaving its pre-training initialisation. Instead, as illustrated in Fig. 10.1, if classes are shown in

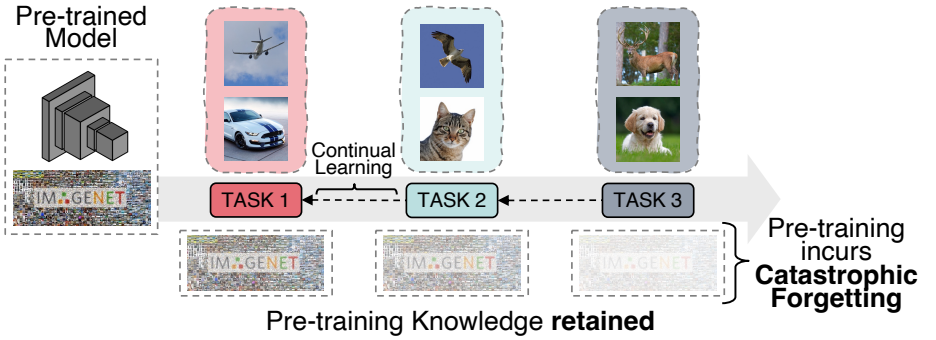


Fig. 10.1: An illustration of the problematic interplay between pre-training and CL: pre-training knowledge is itself subject to catastrophic forgetting while the model is learning from the stream of data; as a consequence, later tasks may not benefit from pre-training features.

a sequential manner, the transfer of pre-training features is only fully available to classes introduced in early tasks; subsequently, pre-training features are swiftly overwritten, barring later tasks from benefiting of pre-training knowledge.

We showcase this phenomenon through a simple preliminary experiment by training ER on S-CIF100 and measuring how much individual ResNet-18 layers differ from their initialisation. A randomly initialised backbone (Fig. 10.2, *left*) significantly alters its parameters at all layers while tasks progress, resulting in a very low Centred Kernel Alignment [82] similarity score already after the first CL task. Similarly, we observe that a backbone pre-trained on Tiny ImageNet (Fig. 10.2, *right*) also undergoes variations in its layers (even though limited, with the exception of the last residual layer).

Such a result indicates that its pre-training parametrisation requires relevant modifications to fit the current training data, leading to the *catastrophic forgetting* of the source pre-training task: namely, the latter is swiftly forgotten as the network focuses on the initial CL tasks. This is validated by the decreasing accuracy for pre-training data of a k-NN classifier trained on *Layer-3* and *-4* representations in Fig. 10.2 (*right*).

To sum up, while pre-training is certainly beneficial, the model inexorably drifts away from it one task after the other. If the first task takes full advantage of it, the optimisation of later tasks instead starts from an initialisation that increasingly differs from the one attained by pre-training. This is detrimental, as classes introduced later might be likewise advantaged by the reuse of different pieces of the initial knowledge. To account for such a disparity and let all tasks profit equally from pre-training, this chapter introduces an *ad-hoc* strategy for fully ex-

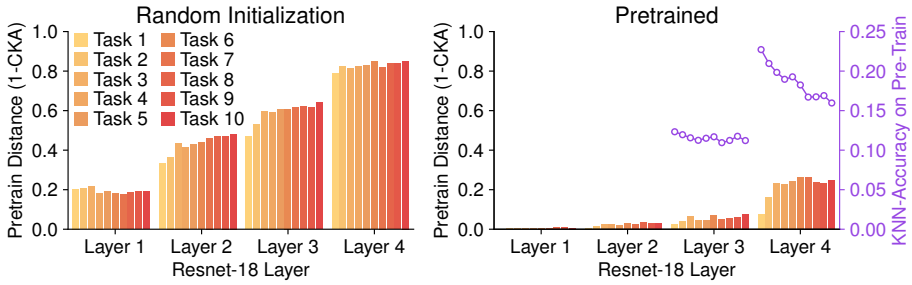


Fig. 10.2: Forgetting of the initialisation, measured as the distance from pre-training (1−CKA [82]) (lower is better) and k-NN accuracy (higher is better). Features extracted by a pre-trained model remain closer to the initialisation w.r.t. a randomly initialised model; the steady decrease in k-NN accuracy as training progresses reveals that features become less specific for past tasks.

ploiting the source knowledge in CL. Our proposal, called **Transfer without Forgetting (TwF)**, exploits per-layer knowledge distillation [59] from a pre-trained sibling network, thus allowing for the continuous propagation of pre-trained representations.

10.2 Transfer without Forgetting

10.2.1 Continuous Feature Transfer

As training progresses, the input stream introduces new classes that might benefit from the adaptation of specific features of the pre-trained model f_{θ^t}. To enable feature transfer without incurring pre-training forgetting, we maintain a separate copy of θ^t (the *sibling* model) and adopt an intermediate-feature knowledge distillation objective [146, 2, 175, 57]. Considering a subset of L layers, we seek to minimise the distance between the activations of the base network $h_\theta^{(l)} \triangleq h_\theta^{(l)}(x)$ and those from its pre-trained sibling $\widehat{h}^{(l)} \triangleq h_{\theta^t}^{(l)}(x)$:

$$\mathop{\mathbb{E}}_{x\sim\mathcal{T}_c}\left[\sum_{l=1}^{L}\left\|h_\theta^{(l)} - \left[\widehat{h}^{(l)}\right]_m^+\right\|_2^2\right], \tag{10.1}$$

where c is the current task and $[\cdot]_m^+$ indicates the application of a margin ReLU activation [57]. The objective outlined in Eq. 10.1 leads the learner to focus on mirroring the internal representations of the pre-trained teacher and maximising transfer. However, this target alone can produce excessive rigidity and prevent the model from fitting current-task data

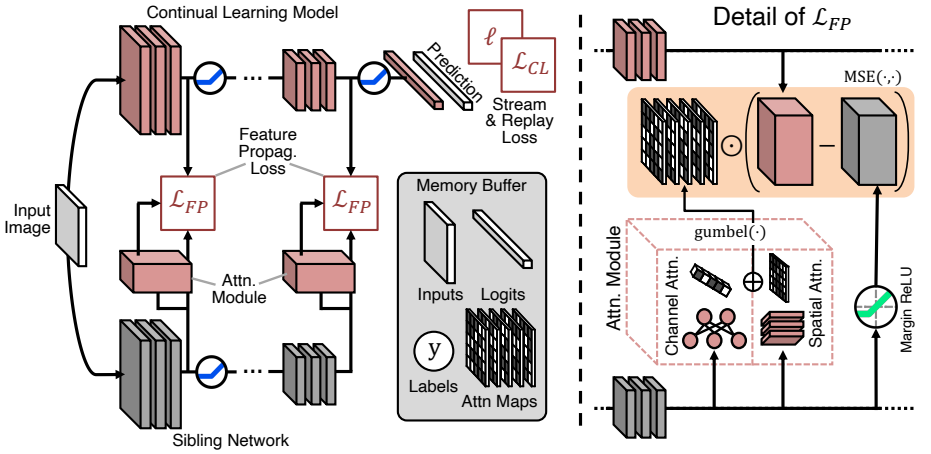


Fig. 10.3: Overview of TwF and detail of $\mathcal{L}_{\text{FP}}$: Given a batch of samples from the current task or from $\mathcal{M}$, we *i)* extract intermediate features from both the student and fixed sibling backbones at multiple layers; *ii)* compute the corresponding binarised attention maps $\mathbb{M}(\cdot)$; *iii)* pull the attention-masked representations of the two models closer.

altogether. Taking inspiration from [175], we thus adopt a weighted version of Eq. 10.1. In particular, we leverage a dedicated learnable module to compute a binary attention map $\mathbb{M}(\cdot)$ over the feature maps of the sibling, with the purpose of selecting which spatial regions have to be aligned. The objective is consequently updated as follows:

$$\mathop{\mathbb{E}}_{x\sim\mathcal{T}_c}\left[\sum_{l=1}^{L}\left\|\mathbb{M}\left(\widehat{h}^{(l)}\right)\odot\left(h_\theta^{(l)}-\left[\widehat{h}^{(l)}\right]_m^+\right)\right\|_2^2\right], \tag{10.2}$$

where $\odot$ indicates the Hadamard product between two equal-sized tensors. The attention maps $\mathbb{M}(\cdot)$ are computed through specific layers, whose architectural design follows the insights provided in [128]. Specifically, they forward the input activation maps into two parallel branches, producing respectively a Channel Attention map $\mathbb{M}_{\text{Ch}}(\cdot)$ and a Spatial Attention map $\mathbb{M}_{\text{Sp}}(\cdot)$. These two intermediate results are summed and then activated through a binary Gumbel-Softmax sampling [63], which allows modelling discrete *on-off* decisions regarding which pieces of information to propagate. In formal terms:

$$\mathbb{M}(\widehat{h}^{(l)}) \triangleq \text{gumbel}\left(\mathbb{M}_{\text{Ch}}(\widehat{h}^{(l)})+\mathbb{M}_{\text{Sp}}(\widehat{h}^{(l)})\right). \tag{10.3}$$

The Spatial Attention $\mathbb{M}_{\text{Sp}}(\widehat{h}^{(l)})$ regulates the propagation of spatially localised information and consists of four convolutional layers [128]:

$$\mathbb{M}_{\text{Sp}}(\widehat{h}^{(l)}) \triangleq \text{C}_{1\times1} \circ \text{C}_{3\times3} \circ \text{C}_{3\times3} \circ \text{C}_{1\times1} \left(\widehat{h}^{(l)}\right), \tag{10.4}$$

where C denotes a sequence of convolutional, batch normalisation, and ReLU activation layers. On the other hand, the Channel Attention $\mathbb{M}_{\text{Ch}}(\widehat{h}^{(l)})$ similarly regulates teaching-signal propagation across the channels of $\widehat{h}^{(l)}$, following the design suggestions proposed in [62]. Formally, we have:

$$\mathbb{M}_{\text{Ch}}(\widehat{h}^{(l)}) \triangleq \tanh\left(\text{BN}(W_1^{\intercal}\widehat{h}^{(l)}_{\text{GAP}})\right) \cdot \sigma\left(\text{BN}\left(W_2^{\intercal}\widehat{h}^{(l)}_{\text{GAP}}\right)\right) + W_3^{\intercal}\widehat{h}^{(l)}_{\text{GAP}}, \tag{10.5}$$

where W_1, W_2 and W_3 are the weights of three parallel fully-connected layers, BN indicates the application of batch normalisation and $\widehat{h}^{(l)}_{\text{GAP}}$ denotes the application of Global Average Pooling (GAP) on top of $\widehat{h}^{(l)}$.

Attention Map Disentanglement

Without a specific loss term disentangling the attention maps, we could obtain degenerate behaviours, where some of the binary gates may end up always being either on or off. While some recent works provide a target expected activation ratio as a countermeasure [1, 151], we encourage the auxiliary modules to assign different propagation gating masks to different examples, the intuition being that each example should determine a distinctive subset of activations to be forwarded from the sibling. For this purpose, we include an auxiliary loss term [117] as follows:

$$\begin{aligned} \mathcal{L}_{\text{AUX}} &\triangleq -\lambda \sum_{l=1}^{L} \mathop{\mathbb{E}}_{x_1,\ldots,x_n\sim\mathcal{T}_c} \left[\sum_{j=1}^{n} \log \frac{e^{g_{ij}^{\intercal} g_{ij}/T}}{\sum_{k=1}^{n} e^{g_{ij}^{\intercal} g_{ik}/T}}\right], \\ g_{ij} &\triangleq \text{NORM}\left(\text{GAP}\left(\mathbb{M}(\widehat{h}^{(l)}(x_j))\right)\right), \end{aligned} \tag{10.6}$$

where n indicates the batch size, NORM a normalisation layer, T a temperature and finally λ is a scalar weighting the contribution of this loss term to the overall objective. In practice, we ask each vector containing channel-wise average activity to have a low dot product with vectors of other examples.

10.2.2 Knowledge Replay

The training objective of Eq. 10.2 is devised to facilitate selective feature transfer between the in-training model and the immutable sibling. To a degree, this is sufficient to achieve basic robustness against catastrophic

forgetting (as we show experimentally in Sec. 10.4.1): preserving pre-training features already serves this purpose. Still, it is advisable to adopt a targeted strategy to prevent the deterioration of knowledge gathered in previous CL tasks. For this reason, we further introduce a reservoir-populated memory buffer $\mathcal{M}$ and apply the replay strategy typical of DER++ (Eq. 4.5).

Since we also wish to avoid catastrophic forgetting w.r.t. the feature propagation formulated in Eq. 10.2, we also extend it to examples in $\mathcal{M}$ and further limit cross-task interference by making all batch normalisation and fully connected layers in Eq. 10.3, 10.4 and 10.5 conditioned [37] w.r.t. the CL task. This implies adding to $\mathcal{M}$ – for each example x – both its task label t and its corresponding set of binary attention maps $m = (m^1, ..., m^l)$ generated at the time of sampling. Eq. 10.2 is finally updated as:

$$\begin{aligned}\mathcal{L}_{\text{FP}} \triangleq & \mathop{\mathbb{E}}_{\substack{(x,t=c)\sim\mathcal{T}_c \\ (x;t)\sim\mathcal{M}}} \left[\sum_{l=1}^{L} \left\| \mathbb{M}(\widehat{h}^{(l)};t) \odot \left(h^{(l)} - \left[\widehat{h}^{(l)} \right]_m^+ \right) \right\|_2^2 \right] \\ & + \mathop{\mathbb{E}}_{\substack{(x,t,m)\sim\mathcal{M} \\ l=1,...,L}} \left[\text{BCE}\left(\mathbb{M}(\widehat{h}^{(l)};t), m^{(l)} \right) \right],\end{aligned} \tag{10.7}$$

where the second term is an additional replay contribution distilling past attention maps, with BCE indicating the binary cross entropy criterion.

10.2.3 Overall objective

Our proposed **Transfer without Forgetting (TwF)** optimises the following auxiliary CL objective as $\mathcal{L}_{\text{R}}$ for Eq. 2.2, also summarised in Fig. 10.3:

$$\mathcal{L}_{\text{TwF}} \triangleq \mathcal{L}_{\text{DER++}} + \mathcal{L}_{\text{FP}} + \mathcal{L}_{\text{AUX}}. \tag{10.8}$$

We remark that: *i)* while TwF requires keeping a copy of the pre-trained model during training, this is not needed at inference time; *ii)* similarly, task labels t are not needed during inference but only while training, which makes TwF capable of operating under all academic CL settings presented in Sec. 2.2.1; *iii)* the addition of t and m in $\mathcal{M}$ induces a limited memory overhead: t can be obtained from the stored labels y for typical classification tasks with a fixed number of classes per task, while m is a set of Boolean maps that is robust to moderate re-scaling. As maps m take discrete binary values, one could theoretically reduce their occupancy by compressing them with lossless algorithms (*e.g.*, Run-Length Encoding [144] or LZ77 [208]).

10.3 Experiments

In this section, we compare TwF against SOTA CL approaches both in the Class-IL and Task-IL settings starting from a pre-trained initialisa-

FAA (FAF)	**S-CIF10** (*pretr. CIFAR-100*)			
Method	**Class-IL**		**Task-IL**	
JT	92.89 (–)		98.38 (–)	
FT	19.76 (98.11)		84.05 (17.75)	
oEWC	26.10 (88.85)		81.84 (19.50)	
LwF	19.80 (97.96)		86.41 (14.35)	
$\lvert\mathcal{M}\rvert$	500	5120	500	5120
ER	67.24 (38.24)	86.27 (13.68)	96.27 (2.23)	97.89 (0.55)
CO^2L	75.47 (21.80)	87.59 (9.61)	96.77 (1.23)	97.82 (0.53)
iCaRL	76.73 (14.70)	77.95 (12.90)	97.25 (0.74)	97.52 (0.15)
DER++	78.42 (20.18)	87.88 (8.02)	94.25 (4.46)	96.42 (1.99)
ER-ACE	77.83 (10.63)	86.20 (5.58)	96.41 (2.11)	97.60 (0.66)
TwF	**83.65** (**11.59**)	**89.55** (**6.85**)	**97.49** (**0.86**)	**98.35** (**0.17**)

Tab. 10.1: FAA and FAF on S-CIF10 w. pre-training on CIFAR-100.

tion. We present experiments on S-CIF10, S-CIF100[1] and S-CUB200; to ensure a fair comparison, all competitors undergo an initial 200-epoch pre-training phase prior to CL on CIFAR-100 [83], Tiny ImageNet [163][2] and ImageNet [39] respectively.

The evaluated competitors include two regularisation approaches (LwF and oEWC) and five RBMs (the ER baseline and four SOTA approaches: CO^2L, iCaRL, DER++ and ER-ACE). As usual, results include the JT upper bound and the FT baseline.

Across the board, non-rehearsal approaches fail to effectively use the features learnt during pre-training. As those methods are not designed to extract and reuse any useful features from the initialisation, the latter is rapidly forgotten, barring any knowledge transfer in later tasks. This is particularly true for oEWC, whose objective proves to be both too strict to effectively learn the current task and insufficient to retain the initialisation. Most notably, on S-CUB200 oEWC performs worse than the FT baseline on both CL settings.

In contrast, those RBMs that feature some form of distillation (DER++ and iCaRL) prove competitive on all benchmarks. In particular, iCaRL is especially effective on S-CIF100, where it attains the second highest FAA even when equipped with a small memory, thanks to its *herding* buffer construction strategy. However, this effect is less pronounced on S-CIF10 and S-CUB200, where pre-training plays a primary role thanks to the similarity of the two distributions for the former and the higher

[1] W.r.t. Tab. 2.3, we apply 0.1 as lr decay factor and 64 as batch size in S-CIF100.

[2] Due to the size mismatch between CIFAR-100 and Tiny ImageNet, we resize samples from the latter to 32×32 during pre-training.

FAA (FAF)	**S-CIF100** (*pretr. Tiny ImageNet*)			
Method	**Class-IL**		**Task-IL**	
JT	75.20 (–)		93.40 (–)	
FT	09.52 (92.31)		73.50 (20.53)	
oEWC	10.95 (81.71)		65.56 (21.33)	
LwF	10.83 (90.87)		86.19 (4.77)	
$\lvert\mathcal{M}\rvert$	500	2000	500	2000
ER	31.30 (65.40)	46.80 (46.95)	85.98 (6.14)	87.59 (4.85)
CO^2L	33.40 (45.21)	50.95 (31.20)	68.51 (21.51)	82.96 (8.53)
iCaRL	56.00 (19.27)	58.10 (16.89)	**89.99** (**2.32**)	90.75 (1.68)
DER++	43.65 (48.72)	58.05 (29.65)	73.86 (20.08)	86.63 (6.86)
ER-ACE	53.38 (21.63)	57.73 (17.12)	87.21 (3.33)	88.46 (2.46)
TwF	**56.83** (**23.89**)	**64.46** (**15.23**)	89.82 (3.06)	**91.11** (**2.24**)

Tab. 10.2: FAA and FAF on S-CIF100 w. pre-training on Tiny ImageNet.

difficulty of the latter. In these settings, we see iCaRL fall short of DER++, which better manages to maintain and reuse the features available from its initialisation. Moreover, we remark that iCaRL and DER++ show a varying Class-IL performance across different tasks, whereas our method is much less sensitive to the specific task at hand.

While effective on the easier S-CIF10 benchmark, CO^2L does not reach satisfactory results on either S-CIF100 or S-CUB200. We ascribe this result to the high sensitivity of this model to the specifics of its training process (*e.g.*, to the applied transforms and the number of epochs required to effectively train the feature extractor with a contrastive loss). While all other competitors employ the same batch size in our experiments, we made an exception for CO^2L and allowed it to use a batch size of 256 to provide a large enough pool of negative samples. Nevertheless, this method only achieves minor improvement w.r.t. non-rehearsal methods for S-CUB200.

Finally, results across all proposed benchmarks indicate that TwF consistently outperforms all competitors, with an average gain of 4.81% for the Class-IL setting and 2.77% for the Task-IL setting, w.r.t. the second-best performer across all datasets (DER++ and ER-ACE, respectively). This effect is especially pronounced for smaller buffers on S-CIF10 and S-CUB200, for which the pre-training provides a more valuable source of knowledge, revealing the efficacy of our proposal to retain and adapt features available from initialisation through distillation. Moreover, its performance gain is consistent over all settings, indicating that the proposed approach can be flexibly applied.

FAA (FAF)	**S-CUB200** (*pretr. ImageNet*)			
Method	**Class-IL**		**Task-IL**	
JT	78.54 (–)		86.48 (–)	
FT	8.56 (82.38)		36.84 (50.95)	
oEWC	8.20 (71.46)		33.94 (40.36)	
LwF	8.59 (82.14)		22.17 (67.08)	
$\|\mathcal{M}\|$	400	1000	400	1000
ER	45.82 (40.76)	59.88 (25.65)	75.26 (9.82)	80.19 (4.52)
CO^2L	8.96 (32.04)	16.53 (20.99)	22.91 (26.42)	35.79 (16.61)
iCaRL	46.55 (12.48)	49.07 (11.24)	68.90 (3.14)	70.57 (3.03)
DER++	56.38 (26.59)	67.35 (13.47)	77.16 (7.74)	82.00 (3.25)
ER-ACE	48.18 (25.79)	58.19 (16.56)	74.34 (9.78)	78.27 (6.09)
TwF	**57.78** (**18.32**)	**68.32** (**6.74**)	**79.35** (**5.77**)	**82.81** (**2.14**)

Tab. 10.3: FAA and FAF on S-CUB200 w. pre-training on ImageNet.

$\mathcal{L}_{\text{DER++}}$	$\mathcal{L}_{\text{FP}}$	$\mathcal{L}_{\text{AUX}}$	**S-CIF10**			**S-CIF100**			**S-CUB200**		
	$\|\mathcal{M}\|$		w/o/buf.	500	5120	w/o/buf.	500	2000	w/o/buf.	400	1000
✓	✓	✓	–	**83.65**	**89.55**	–	**56.83**	**64.46**	–	**59.67**	**68.32**
✓	–	–	–	75.79	87.54	–	44.01	57.84	–	56.53	67.29
✓	✓	–	–	83.29	89.53	–	55.50	63.53	–	59.06	67.83
–	✓	–	60.07	62.63	62.75	49.14	50.20	50.22	37.57	38.43	38.93
–	✓	✓	60.90	63.19	63.79	49.74	50.88	50.52	37.99	39.20	39.31

Tab. 10.4: Impact of each loss term and of using no replay memory on TwF. Results given in the Class-IL scenario following the same experimental settings as Tab.10.1-10.3.

10.4 Analysis

To further characterise the proposed TwF, we present here an additional analysis of the contribution of each loss term, an investigation on alternative approaches for achieving the preservation of pre-training features and a final study on the applicability of TwF when pre-training data is not closely related to CL data.

10.4.1 Breakdown of the Individual Terms of TwF

To better understand the importance of the distinct loss terms in Eq. 10.8 and their connection, we explore their individual contribution to the final accuracy of TwF in Tab. 10.4. Based on these results, we make the

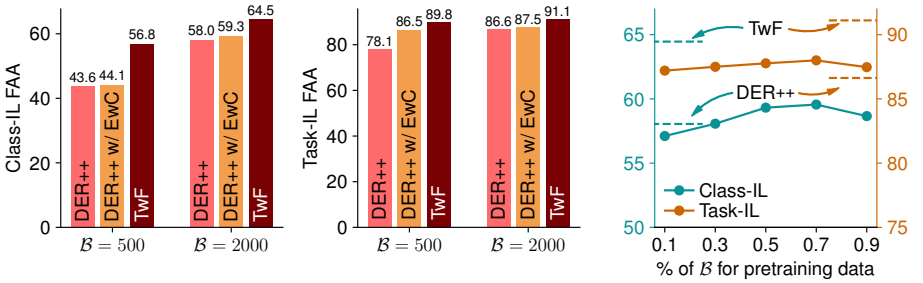


Fig. 10.4: Class-IL (left) and Task-IL (centre) FAA performance comparison of our proposal with different possible methods to retain knowledge from pre-training. (Right) Influence of different allocation rates of pre-training examples in $\mathcal{M}$ for DER++, $|\mathcal{M}| = 2000$.

following observations: *i)* $\mathcal{L}_{\text{DER++}}$ is the most influential loss term and it is indispensable to achieve results in line with the SOTA; *ii)* $\mathcal{L}_{\text{FP}}$ applied on top of $\mathcal{L}_{\text{DER++}}$ induces better handling of pre-training transfer, as testified by the increased accuracy; *iii)* $\mathcal{L}_{\text{AUX}}$ on top of $\mathcal{L}_{\text{FP}}$ reduces activation overlapping and brings a small but consistent improvement.

Further, in the columns labelled as $^{\text{w/o}}/_{\text{buf.}}$, we consider what happens if TwF is provided **no replay example at all** and only optimises $\mathcal{L}_{\text{FP}}$ and $\mathcal{L}_{\text{AUX}}$ on current-task examples. Compared to oEWC in Tab. 10.1-10.3 – the best non-replay method in our experiments – we clearly see preserving pre-training features is in itself a much more effective approach, even with rehearsal out of the picture.

10.4.2 Alternative Pre-Training Preservation Approaches

TwF is designed to both preserve pre-training knowledge and facilitate its transfer. However, other approaches could be envisioned for the same purpose. Hence, we compare here TwF with two alternative baselines for pre-training preservation.

First, we complement a strong approach such as DER++ with an additional regularisation term based on EWC:

$$\mathcal{L}_{\text{EwC}} = \lambda(\theta - \theta^t)^T \,\text{diag}(F)(\theta - \theta^t), \tag{10.9}$$

where $\text{diag}(F)$ indicates the diagonal of the empirical Fisher Information Matrix, estimated on the pre-training data at the optimum θ^t. When equipped with this additional loss term, DER++ is anchored to its initialisation and prevented from changing its pre-training weights significantly, while its replay-based loss term prevents forgetting of knowledge acquired in previous tasks. As shown by Fig. 10.4 (left, centre), the EwC loss

FAA (FAF)	**Class-IL**		**Task-IL**	
$\|\mathcal{M}\|$	500	2000	500	2000
iCaRL	39.59 (21.81)	42.02 (18.78)	78.89 (4.04)	80.65 (2.24)
DER++	36.46 (53.47)	52.29 (24.04)	75.05 (16.22)	83.36 (8.04)
TwF	**43.56** (**40.02**)	**56.15** (**21.51**)	**80.89** (**10.12**)	**87.30** (**3.12**)

Tab. 10.5: Diverse pre-training: FAA on S-CIF100 pre-tr.d on SVHN.

allows DER++ to improve its accuracy on S-CIF100 with Tiny ImageNet pre-training (especially in the Task-IL setting). However, this improvement is not actively promoting feature reuse and thus falls short of TwF. We finally remark that TwF and DER++ w/ EwC have a comparable memory footprint (both retain the initialisation checkpoint).

Differently, we can assume that pre-training data is available and treat it as an auxiliary data stream. To evaluate this strategy with a bounded memory footprint, we test DER++ on S-CIF100 with different percentages of $\mathcal{M}$ dedicated to pre-training images from Tiny ImageNet. The results shown in Fig. 10.4 (right) confirm our claim: DER++ coupled with pre-training rehearsal improves over DER++ with only pre-training. This finding proves that, if pre-training is available, it is beneficial to guard it against catastrophic forgetting. However, by replaying pre-training data, we require the model to maintain its predictive capabilities on the classes of the source task, *i.e.*, we enforce both backward and forward transfer. TwF, instead, allows the model to disregard the classes of the source dataset, as long as the transfer of its internal representations favours the learning of new tasks (*i.e.*, it only enforces **forward transfer**).

10.4.3 Role of Pre-Training Datasets

As a final study, we seek to further test the ability of TwF to adapt features from the pre-training. Specifically, we study a scenario where the source and target data distributions are highly dissimilar: namely, we first pre-train a ResNet18 backbone on SVHN [120] and then follow with S-CIF100. We compare our model with iCaRL and DER++; the results in Tab. 10.5 suggest that our method outranks the competitors not only when pre-trained on a similar dataset – as in Tab. 10.2 – but also when the tasks are very dissimilar. We argue that this result further shows the ability of TwF to identify which pre-training features can be profitably transferred.

10.5 Conclusions

This chapter investigated the interplay between pre-training and CL, highlighting that catastrophic forgetting affects the former as training progresses and hence prevents later tasks from transferring pre-training features. We proposed TwF, a novel CL approach to facilitate the continuous distillation of pre-training features to the online learner and showed that its application results in a clear performance gain over standard CL methods initialised from a pre-trained backbone.

Recently, we see an increasing number of CL works adopting pre-training baselines [181, 64, 179, 178, 162]. This is due to the increasing adoption of large attention-based architectures (typically ViTs [41]), whose operation necessitates large-scale pre-training. The extension of the findings of this chapter to such architectures is an open research direction: our preliminary results indicate that ViTs may be subject to a radically different forgetting that could be possibly linked to the model's lack of inductive biases when compared with CNNs.

Part IV

Conclusion

This thesis focused on classification Continual Learning (CL) problems, focusing in particular on Rehearsal-Based CL Methods (RBMs), which constitute the preferred solution thanks to their ease of use and effectiveness.

After an introductory technical discussion covering the basics of CL, the existing experimental scenarios, benchmarks and approaches (Chap. 2), we presented several proposals for new RBMs: an improved version of the Experience Replay baseline (Chap. 3), a distillation-replay solution allowing for the preservation of secondary information within the model (Dark Experience Replay, Chap. 4) and its extension obtained by a careful management of learning dynamics (eXtended Dark Experience Replay, Chap. 5), a plug-in loss term designed to prevent memory buffer overfitting in RBMs (Lipschitz-Driven Experience Replay, Chap. 6) and a second one aimed at enforcing geometric constraints on the online learner's latent space (Continual Spectral Regulariser, Chap. 7). All proposed approaches are evaluated by experimental means with a comparison with state-of-the-art CL approaches, as well as through additional empirical analyses aiming at characterising their effects on the in-training model.

Subsequently, we presented several proposals for novel experimental settings going beyond the standard benchmarks in literature. This includes the proposal of more realistic and challenging scenarios either by designing an *ad-hoc* experiment characterised by both swift and gradual distribution shifts (General Continual Learning, Chap. 8) or by removing the assumption of full supervision and requiring the online learner to improve starting from unlabelled examples (Continual Semi-Supervised Learning, Chap. 9). Differently, Chap. 10 investigates the interplay between CL and the common practice of pre-training, revealing that the former interferes with the latter and subsequently proposing a novel solution addressing this issue (Transfer without Forgetting).

The presented research has been developed by the candidate throughout the course of his three-year Ph.D. studies; most of the hereby discussed topics have been the object of additional developments in the CL community, as indicated by the individual concluding sections of each chapter. During this period, CL enjoyed a dramatic rise in popularity, growing from a niche topic to a mainstay trend in major Machine Learning conferences and journals. In addition to the development of increasingly more accurate methods, we witness a renewed interest in the experimentation with novel large-scale attention-based architectures. In this respect, the results in this thesis remain open for further validation on these new models, as does our understanding of catastrophic forgetting.

Appendices

Appendix A

List of Publications

The following list of publications includes all conference papers, journal articles and recent pre-prints published during my Ph.D.; the contents and experimental results published in some of these papers have been included in the previous chapters.

[A] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In *Advances in Neural Information Processing Systems*. 2020.

[B] Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara. Rethinking Experience Replay: a Bag of Tricks for Continual Learning. In *International Conference on Pattern Recognition*. 2020.

[C] Giovanni Bellitto, Matteo Pennisi, Simone Palazzo, Lorenzo Bonicelli, Matteo Boschini, Simone Calderara, and Concetto Spampinato. Effects of Auxiliary Knowledge on Continual Learning. In *International Conference on Pattern Recognition*. 2022.

[D] Matteo Boschini, Lorenzo Bonicelli, Angelo Porrello, Giovanni Bellitto, Matteo Pennisi, Simone Palazzo, Concetto Spampinato, and Simone Calderara. Transfer without Forgetting. In *Proceedings of the European Conference on Computer Vision*. 2022.

[E] Matteo Boschini, Pietro Buzzega, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. Continual semi-supervised learning through contrastive interpolation consistency. *Pattern Recognition Letters*, 2022.

[F] Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-Incremental Continual Learning into the eXtended DER-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[G] Lorenzo Bonicelli, Matteo Boschini, Angelo Porrello, Concetto Spampinato, and Simone Calderara. On the Effectiveness of Lipschitz-Driven Rehearsal in Continual Learning. In *Advances in Neural Information Processing Systems*. 2022.

[H] Emanuele Frascaroli, Riccardo Benaglia, Matteo Boschini, Luca Moschella, Cosimo Fiorini, Emanuele Rodolà, and Simone Calderara. CaSpeR: Latent Spectral Regularization for Continual Learning. *arXiv preprint arXiv:2301.03345*, 2023.

Appendix B

Activities carried out during Ph.D.

Teaching activities

- Laboratory assistant for the "Machine Learning and Deep Learning" master course at UNIMORE (Master's Degree in Computer Engineering, A.Y. 2020-2021, 2021-2022, 2022-2023).

- Laboratory assistant for the "Foundations of Computer Science and Laboratory" undergraduate course at UNIMORE (Bachelor's Degree in Electronic Engineering, A.Y. 2020-2021).

- Teacher for the experimental CS curriculum at Liceo Willigelmo (Scientific High-School, S.Y. 2020-2021, 2021-2022).

- Laboratory Teacher at the 1st and 2nd editions of the "School in AI: Deep Learning, Vision and Language for Industry" organised by UNIMORE (February and September 2022).

- Lecturer for the Executive Master in "Intelligenza Artificiale, Machine learning e Deep learning", organised by Fondazione Democenter (March-April 2022).

- Lecturer for the Executive Program in "Corso Teorico e Pratico in Machine Learning e Deep learning", organised by the BI-REX Consortium (September-October 2022).

- Lecturer for the Industrial Training Course in Machine Learning e Deep learning for the employees of CNH industrial (June-July 2022).

Thesis supervision

- October 22, 2020. Lorenzo Bonicelli. Semi-Supervised Continual Learning: avoid catastrophic forgetting with fewer labels. Computer Engineering Master's Degree, UNIMORE.
- April 15, 2021. Federico Marchesi. Variance-Driven scene Clustering for Anomaly Detection. An Anomaly Detection framework on Continual Learning settings. Computer Engineering Master's Degree, UNIMORE.
- December 2, 2021. Lorenzo Schiroli. Evaluation of Continual Learning approaches to mitigate mode collapse in Generative Adversarial Networks. Computer Engineering Master's Degree, UNIMORE.
- December 2, 2021. Aniello Panariello. Detection of anomalous activities on public transport. Computer Engineering Master's Degree, UNIMORE.
- December 2, 2021. Vipul Kumar. Evaluating Catastrophic Forgetting in Unsupervised Learning. Computer Engineering Master's Degree, UNIMORE.
- October 20, 2022. Monica Millunzi. Self-Supervised Continual Learning, avoid catastrophic forgetting with pretext-task. Computer Engineering Master's Degree, UNIMORE.
- October 20, 2022. Martin Menabue. Applicazione di metodi di Continual Learning per mitigare il catastrophic forgetting nella classificazione di immagini con anomalie. Computer Engineering Master's Degree, UNIMORE.

Participation to national and international projects

- "LEGO.AI: LEarning the Geometry of knOwledge in AI systems" – Italian Ministerial grant PRIN 2020 n. 2020TA3K9N.

Reviewing

- International Conference on Learning Representations (ICLR 2021, 2022)
- International Conference on Machine Learning (ICML 2021, 2022)
- Conference on Neural Information Processing Systems (NeurIPS 2021, 2022)
- International Conference on Image Analysis and Processing (ICIAP 2021)

- Conference on Computer Vision and Pattern Recognition (CVPR 2023)
- ACM Transactions on Multimedia Computing Communications and Applications (TOMM)
- IEEE Transactions on Image Processing (TIP)
- IEEE Transactions on Multimedia (TMM)
- IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)
- Transactions on Machine Learning Research (TMLR)
- International Journal of Communication Systems (IJCS)

  I was among the organisers of the Workshop on Novel Benchmarks and Approaches for Real-World Continual Learning (CL4REAL), in conjunction with ICIAP 2021.

Conferences and schools attended

- NeurIPS 2020: 34th Conference on Neural Information Processing Systems. Online. December 6-12, 2020.
- ICPR 2020: 25th International Conference on Pattern Recognition. Online. January 10-15, 2021.
- RegML 2021: Regularization Methods for Machine Learning Summer School. Online. June 21-25, 2021.
- LPS 2022: European Space Agency's 2022 Living Planet Symposium. Bonn, Germany. May 23-27, 2022.
- ECCV 2022: 17th European Conference on Computer Vision. Tel Aviv, Israel. October 23-27, 2022.

Talks given on my research activity

- June 3, 2021. Continual Learning and Knowledge Distillation. BeerAI at Ammagamma, Modena, Italy.
- September 7, 2021. Neural Networks, Continual Learning, Dark Experience Replay and the Mammoth library. Tetra Pak, Modena, Italy.
- September 29, 2022. Class-Incremental Continual Learning into the eXtended DER-verse. ContinualAI (YouTube).

Bibliography

[1] Davide Abati, Jakub Tomczak, Tijmen Blankevoort, Simone Calderara, Rita Cucchiara, and Babak Ehteshami Bejnordi. Conditional channel gated networks for task-aware continual learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[2] Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. Knowledge distillation from internal representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[3] Hongjoon Ahn, Jihwan Kwak, S. Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. SS-IL: Separated Softmax for Incremental Learning. In *IEEE International Conference on Computer Vision*, 2021.

[4] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems*, 2019.

[5] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019.

[6] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient Based Sample Selection for Online Continual Learning. In *Advances in Neural Information Processing Systems*, 2019.

[7] Wangpeng An, Haoqian Wang, Yulun Zhang, and Qionghai Dai. Exponential decay sine wave learning rate for fast deep neural network training. In *IEEE International Conference on Visual Communications and Image Processing*, 2017.

[8] Cem Anil, James Lucas, and Roger Grosse. Sorting out Lipschitz function approximation. In *International Conference on Machine Learning*, 2019.

[9] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *International Conference on Learning Representations Workshop*, 2022.

[10] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *International Conference on Learning Representations Workshop*, 2018.

[11] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *British Machine Vision Conference*, 2016.

[12] Jihwan Bang, Hyunseo Koh, Seulki Park, Hwanjun Song, Jung-Woo Ha, and Jonghyun Choi. Online Continual Learning on a Contaminated Data Stream with Blurry Task Boundaries. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[13] Jean M Barnes and Benton J Underwood. "Fate" of first-list associations in transfer theory. *Journal of Experimental Psychology*, 1959.

[14] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, 2017.

[15] Ari S Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space. In *International Conference on Learning Representations Workshop*, 2019.

[16] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 2019.

[17] Prashant Shivaram Bhat, Bahram Zonooz, and Elahe Arani. Consistency is the key to further mitigating catastrophic forgetting in continual learning. In *Conference on Lifelong Learning Agents*, 2022.

[18] Jochen Bröcker. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society*, 2009.

[19] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New Insights on Reducing Abrupt Representation Change in Online Continual Learning. In *International Conference on Learning Representations Workshop*, 2022.

[20] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

[21] Margaret F Carr, Shantanu P Jadhav, and Loren M Frank. Hippocampal replay in the awake state: a potential substrate for memory consolidation and retrieval. *Nature Neuroscience*, 2011.

[22] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[23] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *IEEE International Conference on Computer Vision*, 2021.

[24] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-SGD: Biasing Gradient Descent Into Wide Valleys. In *International Conference on Learning Representations Workshop*, 2017.

[25] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision*, 2018.

[26] Arslan Chaudhry, Albert Gordo, Puneet Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[27] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM. In *International Conference on Learning Representations Workshop*, 2019.

[28] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. In *International Conference on Machine Learning Workshop*, 2019.

[29] Jeff Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. In *Problems in analysis*. Princeton University Press, 1969.

[30] Patrick H Chen, Wei Wei, Cho-jui Hsieh, and Bo Dai. Overcoming catastrophic forgetting by generative regularization. In *International Conference on Machine Learning*, 2021.

[31] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

[32] Xu Cheng, Zhefan Rao, Yilan Chen, and Quanshi Zhang. Explaining knowledge distillation by quantifying the knowledge. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[33] Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In *International Conference on Machine Learning*, 2020.

[34] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, 2017.

[35] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[36] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *IEEE International Conference on Computer Vision*, 2021.

[37] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, 2017.

[38] Riccardo Del Chiaro, Bartłomiej Twardowski, Andrew Bagdanov, and Joost Van de Weijer. Ratt: Recurrent attention to transient tasks for continual image captioning. In *Advances in Neural Information Processing Systems*, 2020.

[39] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2009.

[40] Mohammad Mahdi Derakhshani, Xiantong Zhen, Ling Shao, and Cees Snoek. Kernel continual learning. In *International Conference on Machine Learning*, 2021.

[41] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations Workshop*, 2021.

[42] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the European Conference on Computer Vision*, 2020.

[43] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *Proceedings of the European Conference on Computer Vision*, 2020.

[44] Sayna Ebrahimi, Suzanne Petryk, Akash Gokul, William Gan, Joseph E Gonzalez, Marcus Rohrbach, and Trevor Darrell. Remembering for the right reasons: Explanations reduce catastrophic forgetting. *Applied AI Letters*, 2021.

[45] Sebastian Farquhar and Yarin Gal. Towards Robust Evaluations of Continual Learning. In *International Conference on Machine Learning Workshop*, 2018.

[46] Enrico Fini, Victor G Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[47] Qiankun Gao, Chen Zhao, Bernard Ghanem, and Jian Zhang. R-dfcil: Relation-guided representation learning for data-free class incremental learning. In *Proceedings of the European Conference on Computer Vision*, 2022.

[48] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2007.

[49] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *International Conference on Learning Representations Workshop*, 2014.

[50] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 2021.

[51] Florian Graf, Sebastian Zeng, Marc Niethammer, and Roland Kwitt. On Measuring Excess Capacity in Neural Networks. In *Advances in Neural Information Processing Systems*, 2022.

[52] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

[53] Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *International Conference on Machine Learning*, 2022.

[54] Mustafa Burak Gurbuz and Constantine Dovrolis. Nispa: Neuro-inspired stability-plasticity adaptation for continual learning in sparse networks. In *International Conference on Machine Learning*, 2022.

[55] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision*, 2017.

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE International Conference on Computer Vision*, 2015.

[57] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *IEEE International Conference on Computer Vision*, 2019.

[58] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Dark knowledge. Technical report, Google inc., 2014.

[59] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *Neural Information Processing Systems Workshops*, 2015.

[60] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[61] Yujia Huang, Huan Zhang, Yuanyuan Shi, J Zico Kolter, and Anima Anandkumar. Training Certifiably Robust Neural Networks with Efficient Local Lipschitz Bounds. In *Advances in Neural Information Processing Systems*, 2021.

[62] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International Conference on Machine Learning*, 2018.

[63] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations Workshop*, 2017.

[64] Paul Janson, Wenxuan Zhang, Rahaf Aljundi, and Mohamed Elhoseiny. A simple baseline that questions the use of pretrained-models in continual learning. In *Neural Information Processing Systems Workshops*, 2022.

[65] Stanislaw Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. In *International Conference on Artificial Neural Networks*, 2018.

[66] Zhong Ji, Jin Li, Qiang Wang, and Zhongfei Zhang. Complementary Calibration: Boosting General Continual Learning with Collaborative Distillation and Self-Supervision. *IEEE Transactions on Image Processing*, 2022.

[67] Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient-based editing of memory examples for online task-free continual learning. In *Advances in Neural Information Processing Systems*, 2021.

[68] Yu Jin, Andreas Loukas, and Joseph JaJa. Graph Coarsening with Preserved Spectral Properties. In *International Conference on Artificial Intelligence and Statistics*, 2020.

[69] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[70] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 2021.

[71] Sandesh Kamath, Amit Despande, and KV Subrahmanyam. On Adversarial Robustness of Small vs Large Batch Training. In *International Conference on Machine Learning Workshop*, 2019.

[72] Zhiqi Kang, Enrico Fini, Moin Nabi, Elisa Ricci, and Karteek Alahari. A soft nearest-neighbor framework for continual semi-supervised learning. *arXiv preprint arXiv:2212.05102*, 2022.

[73] Nazmul Karim, Umar Khalid, Ashkan Esmaeili, and Nazanin Rahnavard. CNLL: A Semi-supervised Approach For Continual Noisy Label Learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[74] Andrej Karpathy. The Tesla Data Engine – Tesla Autonomy Day. Technical report, Tesla, 2019.

[75] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, 2017.

[76] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations Workshop*, 2017.

[77] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*, 2020.

[78] Chris Dongjoo Kim, Jinseo Jeong, and Gunhee Kim. Imbalanced continual learning with partitioning reservoir sampling. In *Proceedings of the European Conference on Computer Vision*, 2020.

[79] Chris Dongjoo Kim, Jinseo Jeong, Sangwoo Moon, and Gunhee Kim. Continual learning on noisy data streams via self-purified replay. In *IEEE International Conference on Computer Vision*, 2021.

[80] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations Workshop*, 2017.

[81] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017.

[82] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, 2019.

[83] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[84] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration. In *Advances in Neural Information Processing Systems*, 2019.

[85] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world. In *International Conference on Learning Representations Workshop*, 2016.

[86] Carlos Lassance, Vincent Gripon, and Antonio Ortega. Representing deep neural networks latent space geometries with graphs. *MDPI Algorithms*, 2021.

[87] Alexis Lechat, Stéphane Herbin, and Frédéric Jurie. Semi-supervised class incremental learning. In *International Conference on Pattern Recognition*, 2021.

[88] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[89] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *International Conference on Machine Learning Workshop*, 2013.

[90] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multi-way Spectral Partitioning and Higher-Order Cheeger Inequalities. *Journal of the ACM*, 2014.

[91] Sungyoon Lee, Jaewook Lee, and Saerom Park. Lipschitz-Certifiable Training with a Tight Outer Bound. In *Advances in Neural Information Processing Systems*, 2020.

[92] Klas Leino, Zifan Wang, and Matt Fredrikson. Globally-robust neural networks. In *International Conference on Machine Learning*, 2021.

[93] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the Loss Landscape of Neural Nets. In *Advances in Neural Information Processing Systems*, 2018.

[94] Jin Li, Zhong Ji, Gang Wang, Qiang Wang, and Feng Gao. Learning from Students: Online Contrastive Distillation Network for General Continual Learning. In *International Joint Conference on Artificial Intelligence*, 2022.

[95] Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. In *International Conference on Learning Representations Workshop*, 2020.

[96] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[97] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 2014.

[98] Hsueh-Ti Derek Liu, Francis Williams, Alec Jacobson, Sanja Fidler, and Or Litany. Learning Smooth Neural Functions via Lipschitz Regularization. In *SIGGRAPH International Conference on Computer Graphics and Interactive Techniques*, 2022.

[99] Huan Liu, Li Gu, Zhixiang Chi, Yang Wang, Yuanhao Yu, Jun Chen, and Jin Tang. Few-shot class-incremental learning via entropy-regularized data-free replay. In *Proceedings of the European Conference on Computer Vision*, 2022.

[100] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the Variance of the Adaptive Learning Rate and Beyond. In *International Conference on Learning Representations Workshop*, 2020.

[101] Shichen Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Generalized zero-shot learning with deep calibration network. In *Advances in Neural Information Processing Systems*, 2018.

[102] Xuan Liu, Xiaoguang Wang, and Stan Matwin. Improving the interpretability of deep neural networks with knowledge distillation. In *IEEE International Conference on Data Mining Workshops*, 2018.

[103] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 2018.

[104] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, 2017.

[105] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, 2017.

[106] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations Workshop*, 2022.

[107] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[108] Davide Maltoni and Vincenzo Lomonaco. Continuous learning in single-incremental-task scenarios. *Neural Networks*, 2019.

[109] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation*, 1989.

[110] Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. An empirical investigation of the role of pre-training in lifelong learning. In *International Conference on Machine Learning*, 2021.

[111] Aditya K Menon, Ankit Singh Rawat, Sashank Reddi, Seungyeon Kim, and Sanjiv Kumar. A statistical perspective on distillation. In *International Conference on Machine Learning*, 2021.

[112] Nicolas Michel, Romain Negrel, Giovanni Chierchia, and Jean-Fmnçois Bercher. Contrastive Learning for Online Semi-Supervised General Continual Learning. In *IEEE International Conference on Image Processing*, 2022.

[113] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the Role of Training Regimes in Continual Learning. In *Advances in Neural Information Processing Systems*, 2020.

[114] Sudhanshu Mittal, Silvio Galesso, and Thomas Brox. Essentials for Class Incremental Learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[115] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations Workshop*, 2018.

[116] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[117] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. Subclass distillation. *arXiv preprint arXiv:2002.03936*, 2020.

[118] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

[119] Yuji Nakatsukasa and Nicholas J Higham. Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the SVD. *SIAM Journal on Scientific Computing*, 2013.

[120] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems*, 2011.

[121] Behnam Neyshabur, Srinadh Bhojanapalli, David Mcallester, and Nati Srebro. Exploring Generalization in Deep Learning. In *Advances in Neural Information Processing Systems*, 2017.

[122] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2018.

[123] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, 2018.

[124] Chapelle Olivier, Scholkopf Bernhard, and Zien Alexander. *Semi-supervised learning*. MIT Press, 2006.

[125] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)*, 2012.

[126] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 2009.

[127] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.

[128] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Bam: Bottleneck attention module. In *British Machine Vision Conference*, 2018.

[129] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer Vision*, 2019.

[130] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[131] Federico Pernici, Matteo Bruni, Claudio Baecchi, and Alberto Del Bimbo. Regular polytope networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[132] Federico Pernici, Matteo Bruni, Claudio Baecchi, Francesco Turchini, and Alberto Del Bimbo. Class-incremental learning with pre-allocated fixed classifiers. In *International Conference on Pattern Recognition*, 2021.

[133] Quang Pham, Chenghao Liu, and Steven Hoi. DualNet: Continual Learning, Fast and Slow. In *Advances in Neural Information Processing Systems*, 2021.

[134] Julien Pourcel, Ngoc-Son Vu, and Robert M French. Online task-free continual learning with dynamic sparse distributed memory. In *Proceedings of the European Conference on Computer Vision*, 2022.

[135] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. GDumb: A simple approach that questions our progress in continual learning. In *Proceedings of the European Conference on Computer Vision*, 2020.

[136] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial Robustness through Local Linearization. In *Advances in Neural Information Processing Systems*, 2019.

[137] Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological Review*, 1990.

[138] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[139] Tamar Reitich-Stolero and Rony Paz. Affective memory rehearsal with temporal sequences in amygdala neurons. *Nature Neuroscience*, 2019.

[140] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[141] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference. In *International Conference on Learning Representations Workshop*, 2019.

[142] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems*, 2018.

[143] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 1995.

[144] A Harry Robinson and Colin Cherry. Results of a prototype television bandwidth compression scheme. In *Proceedings of the IEEE*, 1967.

[145] Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer Graphics Forum*, 2017.

[146] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations Workshop*, 2015.

[147] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[148] Joao Paulo Schwarz Schuler. CAI neural API. `https://github.com/joaopauloschuler/neural-api`, 2019.

[149] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, 2018.

[150] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 2019.

[151] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming Catastrophic Forgetting with Hard Attention to the Task. In *International Conference on Machine Learning*, 2018.

[152] Amazon Web Services. Renate: Automatic Neural Networks Retraining and Continual Learning in Python, 2022. `https://github.com/awslabs/Renate`.

[153] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016.

[154] Yuzhang Shang, Bin Duan, Ziliang Zong, Liqiang Nie, and Yan Yan. Lipschitz continuity guided knowledge distillation. In *IEEE International Conference on Computer Vision*, 2021.

[155] Hang Shao, Abhishek Kumar, and P. Thomas Fletcher. The Riemannian Geometry of Deep Generative Models. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[156] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[157] Hyounguk Shon, Janghyeon Lee, Seung Hwan Kim, and Junmo Kim. Dlcft: Deep linear continual fine-tuning for general incremental learning. In *Proceedings of the European Conference on Computer Vision*, 2022.

[158] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016.

[159] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and Computation*, 1989.

[160] James Smith, Jonathan Balloch, Yen-Chang Hsu, and Zsolt Kira. Memory-efficient semi-supervised continual learning: The world is its own replay buffer. In *International Joint Conference on Neural Networks*, 2021.

[161] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[162] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. *arXiv preprint arXiv:2211.13218*, 2022.

[163] Stanford. Tiny ImageNet Challenge (CS231n), 2015. `https://www.kaggle.com/c/tiny-imagenet`.

[164] David Stutz, Matthias Hein, and Bernt Schiele. Relating adversarially robust generalization to flat minima. In *IEEE International Conference on Computer Vision*, 2021.

[165] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations Workshop*, 2014.

[166] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, 2017.

[167] Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *Advances in Neural Information Processing Systems*, 2018.

[168] Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 2022.

[169] Eli Verwimp, Matthias De Lange, and Tinne Tuytelaars. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *IEEE International Conference on Computer Vision*, 2021.

[170] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 2019.

[171] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

[172] Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, 2018.

[173] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 1985.

[174] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011.

[175] Kafeng Wang, Xitong Gao, Yiren Zhao, Xingjian Li, Dejing Dou, and Cheng-Zhong Xu. Pay attention to features, transfer learn faster CNNs. In *International Conference on Learning Representations Workshop*, 2019.

[176] Qiang Wang, Jiayi Liu, Zhong Ji, Yanwei Pang, and Zhongfei Zhang. Hierarchical Correlations Replay for Continual Learning. *Knowledge-Based Systems*, page 109052, 2022.

[177] Zhenyi Wang, Li Shen, Le Fang, Qiuling Suo, Tiehang Duan, and Mingchen Gao. Improving task-free continual learning by distributionally robust memory evolution. In *International Conference on Machine Learning*, 2022.

[178] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Proceedings of the European Conference on Computer Vision*, 2022.

[179] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[180] Sebastien C Wong, Adam Gatt, Victor Stamatescu, and Mark D McDonnell. Understanding data augmentation for classification: when to warp? In *International Conference on Digital Image Computing: Techniques and Applications*, 2016.

[181] Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan-Fang Li, Guilin Qi, and Gholamreza Haffari. Pretrained language model in continual learning: A comparative study. In *International Conference on Learning Representations Workshop*, 2021.

[182] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[183] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[184] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[185] Huan Xu and Shie Mannor. Robustness and generalization. *Machine Learning*, 2012.

[186] Jia Xu, Yiming Li, Yong Jiang, and Shu-Tao Xia. Adversarial defense via local flatness regularization. In *IEEE International Conference on Image Processing*, 2020.

[187] Qingsen Yan, Dong Gong, Yuhang Liu, Anton van den Hengel, and Javen Qinfeng Shi. Learning Bayesian Sparse Networks with Full Experience Replay for Continual Learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[188] Chenglin Yang, Lingxi Xie, Siyuan Qiao, and Alan L Yuille. Training deep neural networks in generations: A more tolerant teacher educates better students. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[189] Guanglei Yang, Enrico Fini, Dan Xu, Paolo Rota, Mingli Ding, Moin Nabi, Xavier Alameda-Pineda, and Elisa Ricci. Uncertainty-aware contrastive distillation for incremental semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[190] Zhewei Yao, Amir Gholami, Qi Lei, Kurt Keutzer, and Michael W Mahoney. Hessian-based analysis of large batch training and robustness to adversaries. *Advances in Neural Information Processing Systems*, 2018.

[191] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1995.

[192] Dong Yin, Mehrdad Farajtabar, Ang Li, Nir Levine, and Alex Mott. Optimization and generalization of regularization-based continual learning: a loss approximation viewpoint. In *International Conference on Machine Learning Workshop*, 2020.

[193] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[194] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

[195] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, 2014.

[196] Fuxun Yu, Zhuwei Qin, Chenchen Liu, Liang Zhao, Yanzhi Wang, and Xiang Chen. Interpreting and evaluating neural network robustness. In *International Joint Conference on Artificial Intelligence*, 2019.

[197] Longhui Yu, Tianyang Hu, Lanqing Hong, Zhen Liu, Adrian Weller, and Weiyang Liu. Continual Learning by Modeling Intra-Class Variation. Technical report, University of Cambridge, 2022.

[198] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, 2021.

[199] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 2017.

[200] Mengyao Zhai, Lei Chen, Jiawei He, Megha Nawhal, Frederick Tung, and Greg Mori. Piggyback GAN: Efficient Lifelong Learning for Image Conditioned Generation. In *Proceedings of the European Conference on Computer Vision*, 2020.

[201] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations Workshop*, 2018.

[202] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 2018.

[203] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. Class-incremental learning via deep model consolidation. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020.

[204] L Zhang, C Bao, and K Ma. Self-Distillation: Towards Efficient and Compact Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[205] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *IEEE International Conference on Computer Vision*, 2019.

[206] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[207] Wang Zhou, Shiyu Chang, Norma Sosa, Hendrik Hamann, and David Cox. Lifelong Object Detection. *arXiv:2009.01129*, 2020.

[208] Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, 1977.