UNIVERSITÀ degli STUDI di CATANIA

DIPARTIMENTO DI INGEGNERIA ELETTRICA, ELETTRONICA E INFORMATICA

PH.D. PROGRAM IN SYSTEMS, ENERGETICS, COMPUTER AND TELECOMMUNICATIONS ENGINEERING
XXXVI CYCLE

Ph.D. Thesis

# NEUROCOGNITIVE-INSPIRED PARADIGMS FOR CONTINUAL LEARNING

GIOVANNI BELLITTO

Ph.D. Coordinator:
Prof. P. ARENA

Supervisor:
Prof. C. SPAMPINATO
Co-Supervisor:
Prof. S. CALDERARA

*To Ferderica, my true beloved, always stood by my side,*
*for bearing with me and giving me the strength to go forward.*
*Always. Together.*
*To my family, with their unwavering support over the years,*
*for always believing in me, I wouldn't be here without their sacrifices.*
*To Concetto, our guide and mentor, never just a supervisor,*
*immensely grateful for having seen something I wasn't even aware of.*
*To Simone, a source of inspiration to us all,*
*for his continuous support and his valuable role in our achievements.*
*To the members of the PeRCeiVe Lab, where I've spent memorable years,*
*for sharing the daily ups and downs of this extraordinary journey.*

*From the depth of my soul,*
*Thank You.*

# ABSTRACT

Emulating human learning is a fundamental component in research towards artificial intelligence (AI).

However, despite the substantial progress in the last decades, humans continue to outperform machines in many visual tasks. The motivation for this discrepancy might be rooted in the lack of a fully understanding the human learning process that is featured by being resilient to task and data changes and keep increasing over time. In contrast, Artificial Neural Networks (ANN) are highly susceptible to shifts in data distribution over time, a shortcoming that hinders the development of intelligent agents that can rapidly adapt to different context and experiences.

Continual Learning (CL) is a paradigm in AI that focuses on the ability of models to learn continuously over time, assimilating new knowledge while concurrently preserving and building upon previously acquired insights. Traditional AI models, when exposed to new data or tasks, often suffer from "Catastrophic Forgetting", where the introduction of new information can overwrite previously learned knowledge, or even erase it. The essence of CL is to counteract this limitation, pioneering algorithms and strategies that empower models to seamlessly integrate new information

without compromising the integrity of their existing knowledge base.

Drawing inspiration from the human cognitive system's remarkable ability to learn, adapt and remember over decades, in this thesis we aim to propose new solutions for AI systems that reflect this adaptability and long-term retention. The ambition is to help usher in a new era of AI where systems not only evolve in response to changing data landscape but also become repositories of accumulated knowledge over extended periods.

We propose to address the problem of Forgetting from two perspectives. In the first part, we design new methods inspired by the human ability to draw on existing knowledge to address new challenges and devise effective solutions. Past experience serves as a valuable reservoir of insights that can be leveraged when tackling new problems. We emulate prior knowledge within a neural network by employing an auxiliary stream of data, that may encompass relevant features for both the current and subsequent tasks. Alternatively, we introduce an hybrid transfer learning approach based on a fixed pre-trained sibling network, which propagates the knowledge inherent in the source domain throughout the continual learning process. Then, we present an efficient strategy for coupling the primary classification task with an orthogonal task that guides training, yielding additional useful knowledge without the need to use external auxiliary data.

In the second part of this dissertation, we present two innovative solutions, deeply inspired by cognitive theories, that attempt to replicate in artificial networks some fundamental human cognitive processes. The first approach exploits the mechanism of human visual system, showing the remarkable property of selective attention to be resistant to forgetting. This inherent robustness of the saliency prediction task, perfectly suits with the continual learning context, improving the performance of a continual classifier. Finally, we introduce a novel *wake-sleep* learning framework, where

the phase of acquiring new knowledge from the current task (wake) alternates with a phase dedicated to consolidating and preparing for subsequent experiences (sleep). This emulation mirrors the role of dreaming in easing the learning process and enhancing the generalization capability.

# CONTENTS

# Part I

# INTRODUCTION

*"Without the ability to accumulate the learned knowledge and use it to learn more knowledge incrementally, a system will probably never be truly intelligent."*

Bing Liu, *Lifelong Machine Learning*, 2017

CHAPTER

# ONE

# OVERVIEW

## 1.1 Introduction

From the moment of birth, and even before, biological organisms are capable to continuously absorb, adapt and evolve based on their interaction with their surroundings. Neuroscientists and biologists, in their quest to unravel the mysteries of the brain, have proposed numerous theories to explain this inexhaustible capacity for learning. In parallel, Machine Learning (ML) researchers have attempted to emulate such organic learning process within Artificial Neural Networks (ANN), with varying degrees of success. Ideally, neural networks should reflect the same ability to learn on a continual basis. However, the journey has not been straightforward.

## 1.2 The World is not stationary

In contrast to the common learning process observed in nature, the predominant approach in ML is *Isolated Learning*. In this approach, once a neural network is defined, it is assumed that all training data is available since the beginning; the model is then trained on this data and subsequently applied to real-world tasks, where it is expected to perform. This static approach overlooks a critical aspect of learning: the dynamic nature of information. In real-world scenarios, information is not always available in its entirety at the outset. Data may become available only later, requiring neural networks to adapt and expand the knowledge sequentially.

Isolated Learning does not account for the retention and accumulation of knowledge. Humans, in contrast, never learn in isolation. We consistently retain past knowledge, leveraging it to facilitate future learning and problem-solving. When faced with a new problem, it is rarely completely new to us: often we recognize parts of it from past experiences or different contexts and we use them as starting points for understanding. This cognitive richness underscores a pivotal aspect of human learning: we never truly start from scratch.

The human mind is never a *blank sheet*. Even as newborns, humans possess a genetically inherited knowledge base – a set of instincts, reflexes, and basic cognitive structures – that serves as the foundation for all subsequent learning. It is dynamic, evolving with layers of memories, experiences, and skills, enabling us to navigate an ever-changing world and tackle more and more complex challenges.

While Isolated Learning has demonstrated to be effective for neural networks, its efficacy heavily relies on the availability of huge amounts of training samples. On the other hand, humans can learn effectively with few

examples. This effectiveness stems from the vast reservoir of accumulated knowledge from the past, allowing us to grasp new concepts with limited data or effort. For instance, assuming we already have embedded knowledge of what a *horse* is, we do not need thousands of samples of horses and zebras to distinguish between them; our prior knowledge provides the necessary context.

Unlike the human brain, which seamlessly integrates new knowledge with old, for neural networks it is extremely hard to emulate this behaviour: they struggle with accumulating knowledge incrementally. When exposed to new data or concepts, they tend to adapt to the new data distribution, often at the expense of previously acquired knowledge. This phenomenon is also known as ”Catastrophic Forgetting”.

Catastrophic forgetting is more than just a minor hiccup in the journey of machine learning; it’s a fundamental roadblock. It challenges the very essence of creating models that can learn and adapt over time, much like humans. Addressing this challenge bring us to the paradigm of Continual Learning (CL). At its core, CL studies the problem of enabling ANNs to learn continuously, acquiring new knowledge while retaining and building on previously learned information. It seeks to emulate the human ability to learn from sequential experiences without the need for revisiting old data constantly. As an ultimate goal, CL aims to find solutions to mitigate the forgetting problem.

## 1.3 Catastrophic Forgetting: a persistent challenge in Machine Learning

Catastrophic Forgetting is a pervasive issue that potentially affects any neural network. Regardless of complexity, when a neural network is sequentially trained on multiple tasks, the weights and the biases of the network – which hold the learned knowledge – are updated to reflect the new information. However, without a specific mechanism to preserve the old knowledge, these updates can interfere or disrupt the representations of the previous tasks.

The phenomenon of catastrophic forgetting has been recognized and examined since the dawn of ML. In the 1980s, researches identified and described it as a potential fundamental limitation of what *distributed architectures* (referred to as the early name of multi-layer perceptron precursors) could do [1, 2, 3, 4, 5, 6]. McCloskey and Cohen [5] observed that under certain conditions, the process of learning new patters partially o completely erase the knowledge that the network had already learned. They termed this event as *"catastrophic interference"* (now replaced by the more common term *forgetting*) and their investigations concluded that a significant part of the issue was due to the fact that neural networks are organized as *one* single set of shared weights, which allow them remarkable abilities to generalize, but at the same time to progressively degrade. Ratcliff [6] conducted extensive experiments examining various alternative ways of updating network weights (e.g., modifying all weights; adjusting only a subset of them; incorporating new hidden units; adding hidden units while changing only connections to and from the new units) discovering that none of the tested methods produced satisfactory results for the entire

test sequence samples. Furthermore, he noted that all the considered cases always led to two alternative behaviours: the last test samples were better identified at the expenses of the early samples, or vice-versa, indicating an inevitable trade-off between recent and older memories. Ratcliff's findings were a concrete illustration of a very complex problem concerning all neural networks trained with gradient-based optimization methods, known as the *stability-plasticity dilemma* [7]. The dilemma revolves around the two opposing needs of neural networks: the requirement of *plasticity* to acquire new data, and the need to *stability* to retain prior knowledge. Excessive plasticity results in high forgetting as old knowledge makes room for the new one. Conversely, excessive stability causes the network to be resistant to learning new experiences. Managing this delicate balance is critical to the development of robust and versatile learning systems.

CHAPTER

# TWO

# BACKGROUND

While the implications of Continual Learning traverse various domains, this thesis specifically focuses on the problem of continual learning for image classification. Image classification is one of the most pivotal problems in the field of computer vision; indeed, the task of correctly categorizing images is not only integral to diverse applications such as medical diagnosis, autonomous driving and content recommendation, but it also serves as a benchmark for assessing the robustness and adaptability of AI systems.

## 2.1 Continual Learning: Formal Definition

CL is the problem of learning from a non-*i.i.d.* (independent and identically distributed) stream of data, with no or limited access to old training samples. More formally, let $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$ be a sequence of data streams, where each pair $(\mathbf{x}, y) \sim \mathcal{D}_i$ denotes a data point $\mathbf{x} \in \mathcal{X}$ with the corresponding class label $y \in \mathcal{Y}$; the sample distributions (in terms of both the

data point distribution and the class label distribution) of different $\mathcal{D}_i$ and $\mathcal{D}_j$ may vary — for instance, class labels from $\mathcal{D}_i$ might be different from those from $\mathcal{D}_j$. In other words, although the data point within $\mathcal{D}_i$ are *i.i.d*, the global $\mathcal{D}$ deviates from this assumption. Given a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$, parameterized by $\boldsymbol{\theta}$, the objective of CL is to train $f$ on $\mathcal{D}$, organized as a sequence of $T$ tasks $\{\tau_1, \ldots, \tau_T\}$, under the constraint that, at a generic task $\tau_i$, the model receives inputs sampled from the corresponding data distribution only, i.e., $(\mathbf{x}, y) \sim D_i$.

The training objective is to optimize a classification loss over the sequence of tasks (without losing accuracy on past tasks) by the model instance at the end of training:

$$\arg\min_{\boldsymbol{\theta}_T} \sum_{i=1}^{T} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i} \Big[ \mathcal{L}\Big( f\left(\mathbf{x}; \boldsymbol{\theta}_T\right), y \Big) \Big] \tag{2.1}$$

where $\mathcal{L}$ is a generic classification loss (e.g., cross-entropy).

## 2.2 Scenarios

Given the complexity of learning over a series of tasks, and the underlying need to find effective solutions, CL has received considerable attentions from researches. In recent years a plethora of methods have been proposed to alleviate the problem of forgetting. However, a multitude of subtle, but crucial differences between evaluation protocol made systematic comparison of early CL methods extremely complex, even among those using the same datasets. The need for a uniform categorization, trying to structure the CL problem, and identifying common frameworks widely shared by the community became imperative. With this aim, three fundamental types, or

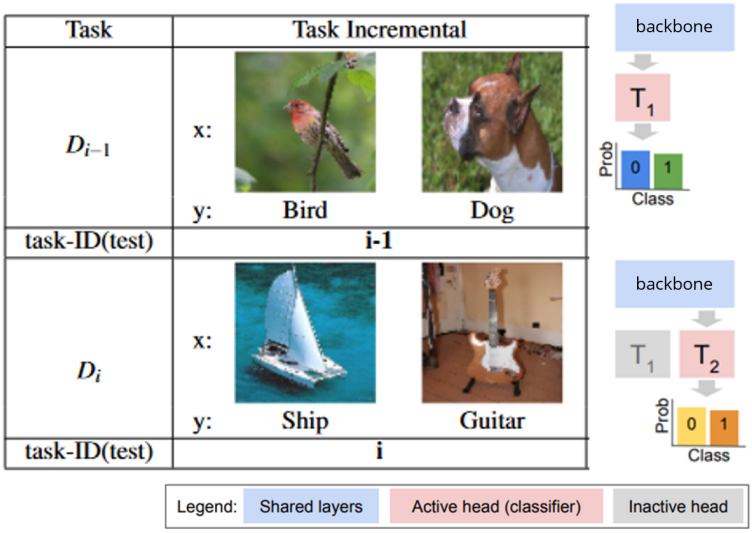



***Figure 2.1:*** *Example of Task Incremental Learning (Task-IL). (x, y and task-ID) represents (input image, label, task identity). In Task-IL, task-ID is explicitly provided. A typical network has a* "multi-headed" *output layer: each task has its own output units, while the rest of the net is shared between tasks.*

scenarios, of CL were identified in order to categorize the most of the proposed methods. Essentially, categorization is based on the availability of the task identity at inference time, and if not, if the model is required to explicitly identify the task [8].

### 2.2.1 Task-Incremental Learning

Task-Incremental Learning (or *Task-IL*) is the case where a neural network is designed to learn multiple distinct sets of classes incrementally (i.e., $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset, \forall i, j \in \{0, \ldots, T\}$ with $T$ the number of tasks). Additionally,

the model explicitly receives the information about the task to solve $t$, as input, also at inference time. Since the model is always informed about the task to tackle, and thus the dimension of the problem is simplified to distinguish between the classes of the task at hand, this is considered the easiest CL scenario. It is also possible to design models equipped with task-specific components. A typical neural network for this scenario has a common backbone net shared for all the tasks, and a multi-headed output layer; at inference time, only the $t$-th head is activated to make predictions.

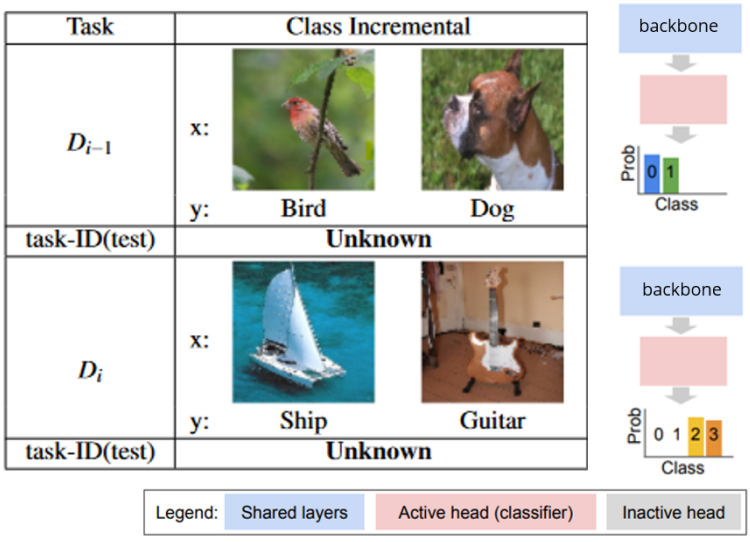



***Figure 2.2:*** *Example of Class Incremental Learning (Class-IL). (x, y and task-ID) represents (input image, label, task identity). In Class-IL, task-ID is not available at testing time. The model must incrementally learn to discriminate between a growing number of classes.*

### 2.2.2 Class-Incremental Learning

In the Class-Incremental Learning (or *Class-IL*) scenario, similarly to Task-IL, each task consists of a unique set of classes, but at inference time the identity of the task to be solved is not available. This makes the classification problem considerably more challenging. In practice, the neural network has to discriminate between an increasing number of classes as the number of tasks grows. The main difficulty lies in distinguishing between classes belonging to different tasks, since the network tends to predict better those of the last task. Due to its intricate nature, Class-IL is considered the most complex among the three CL scenarios. For the same reason, Class-IL is the most popular benchmark when a new CL method is proposed.

### 2.2.3 Domain-Incremental Learning

In the Domain-Incremental Learning (or *Domain-IL*) scenario, the complexity emerges due to varying domain context, while the class labels remain invariant. The overall class number is fixed since the beginning: each task yields the same possible outputs, and at inference time the task identity is unknown. The challenge arises from shifts in class distribution, resulting in different internal representation with each task. Such domain shifts may be due, for example, to different permutations within tasks. Designed specifically for scenarios that require domain adaptation [9, 10, 11], the objective of Domain-IL is to maintain proficient performance levels on previous tasks while utilizing one single model. Practical examples of this CL scenario are recognizing objects under different lighting conditions (e.g., the first task involves identifying items indoors, while the second

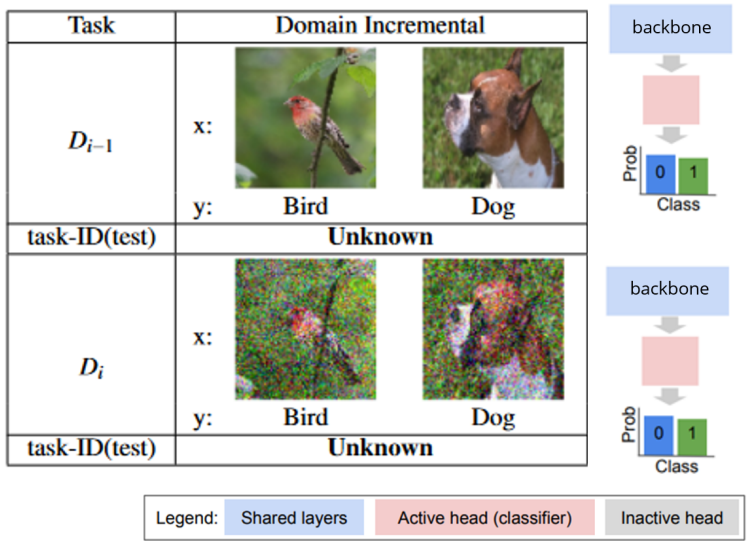



***Figure 2.3:*** *Example of Domain Incremental Learning (Domain-IL). (x, y and task-ID) represents (input image, label, task identity). In Domain-IL, task-ID is not available at testing time. In Domain-IL, each task has the same possible outputs, since the same classes are used in each task, but the in-data distribution changes at each task.*

task focuses on the outdoors), or recognizing traffic signs under distinct weather conditions [12]. Despite the change in environmental context, the categories remain the same, but their appearance and representation might vary significantly, thereby requiring the model to adapt without forgetting its prior learning.

### 2.2.4 A more complex setting: Online Continual Learning

The three CL scenarios just presented, also known as *academic scenarios*, have received significant attention from the research community. However, since they represent abstractions of diverse real-world applications, they also come with some simplifications. One of them concerns the number of training iterations per task. The academic scenarios do not impose any limitation on the number of times an individual data-item can be shown to the model within the boundaries of a given task. Nevertheless, some real cases may impose more severe restrictions on the data availability. Online Continual Learning (OCL), addresses situations where data-item are received in a temporal sequence and must be processed immediately. It is generally based on either Class-IL (oCIL), or Task-IL (oTIL), imposing the constraint that the model cannot review past samples or store them for future reference due to constraints such as memory limits or real-time processing. The problem of OCL will be faced in Chapters 5 and 6.

## 2.3 Benchmarks

To benchmark and evaluate CL strategies, researchers have predominantly relied on common datasets originally designed for supervised image classification, but appropriately re-organized for a context of incremental learning. Here we provide a list of the dataset that will be used in the following chapters:

- **Sequential CIFAR-10 (Seq-CIFAR-10)**: derived from the CIFAR-10 dataset [13], which offers 60,000 32x32 RGB images across 10

classes. It is split into 5 binary tasks, with 5,000 and 1,000 images for training and testing, respectively, per task.

- **Sequential CIFAR-100 (Seq-CIFAR-100)**: it is obtained by dividing the original 100 classes of the CIFAR-100 dataset [13] into 10 consecutive tasks, with 20 classes each organized into 20 classes with 500 32x32 color images per class for training, 100 for testing.

- **Sequential Mini-ImageNet (Seq-Mini-ImageNet)**: it includes a subset of 100 classes from the popular ImageNet dataset [14], devised in a sequence of 20 tasks. Each tasks includes 84x84 RGB images from 5 different classes; for each class, 500 images are used in training and 100 for evaluation.

- **Sequential Tiny-ImageNet (Seq-Tiny-ImageNet)**: it is obtained by splitting the Tiny ImageNet dataset [15] into 10 tasks with 20 classes each. Images are reshaped to 64x64, and for each class there are 500 images for training and 50 images for testing.

- **Sequential Micro-ImageNet (Seq-Micro-ImageNet)**: designed as a subset of Seq-Tiny-ImageNet, it consists of 20 classes, split into 5 tasks of 4 classes each.

- **Sequential FG-ImageNet (Seq-FC-ImageNet)**: it is a fine-grained image classification benchmark with 100 classes of animals extracted from ImageNet [14], used to test CL methods on a more challenging task. Each class contains 500 samples for training and 50 for evaluation[1].

[1]FC-ImageNet is derived from https://www.kaggle.com/datasets/ambityga/imagenet100

- **Sequential CUB-200 (Seq-CUB-200)**: it derives from Caltech-UCSD Birds-200 datasets [16], split into 10 tasks of 20 classes each, with just around 30 training and testing images per class resized to 224x224.

## 2.4 Metrics

Here we list the main evaluation metrics proposed for CL in literature:

- **Final Average Accuracy (FAA)** is the final average accuracy of the model after learning the last task $T$, defined as:

$$FAA \triangleq \frac{1}{T}\sum_{i=1}^{T} a_i^T. \tag{2.2}$$

  where $a_i^t$ is the model accuracy on the $i$-th task after training on task $t$. FAA is the primary evaluation metric, used in every CL manuscripts.

- **Final Backward Transfer (FBWT)** [17] measures how learning the current task $t$ affects on the performance on a previous task $k < t$. There is a positive backward transfer when learning about task $t$ increases the performance on some preceding task $k$, while negative backward transfer suggest the opposite:

$$FBWT \triangleq \frac{1}{T-1}\sum_{i=1}^{T-1} a_i^T - a_i^i \tag{2.3}$$

- **Final Forward Transfer (FFWT)** measure the influence that learning the current task $t$ has in the performance on a future task $k > t$.

Positive forward transfer implies that the knowledge from prior tasks have enhanced the learning of a new task. FFWT is computed as the difference between the accuracy just before starting training on a given task and the one of the random-initialized network, averaged across all tasks:

$$FFWT \triangleq \frac{1}{T-1} \sum_{i=2}^{T} a_i^{i-1} - a_i^{random} \tag{2.4}$$

where $a_i^{random}$ is the accuracy of the network with random initialization on the $i$-th task.

For all these metrics, the larger these values, the better the model. In the *ideal* scenario, we would observe:

– Positive Forward Transfer: this means that learning earlier tasks makes it easier to learn subsequent task due to the shared knowledge or generalizable features.

– No negative Backward Transfer, i.e., when new tasks are learned, the performance on previously learned tasks remains unaffected.

The typical scenario encountered is, instead, as follows:

– Backward Transfer is always *negative*, as a direct consequence of catastrophic forgetting.

– In terms of Forward Transfer, in the Class-IL and Task-IL scenarios, where distinct classes are learned in distinct tasks, a positive transfer is essentially impossible (the model should be capable of correctly classifying unseen classes); in Domain-IL setting, where original images are provided with different transformation, positive values might be feasible.

Finally, forgetting, as a measure of the severity of performance degradation, can be quantified as follow:

- **Final Forgetting (FF)** measures the average performance degradation occurring on past tasks between their best values and the current accuracy:

$$FF \triangleq \frac{1}{T-1} \sum_{i=1}^{T-1} \max_{l \in \{1,\dots,T-1\}} a_i^l - a_i^T \tag{2.5}$$

If FF is a positive value, it indicates that the model has forgotten some of what it learned initially. If negative, it implies that subsequent training improved the model performance on the original task.

Other possible (secondary) criteria in order to evaluate the model performance may be:

- **Memory consumption**: the amount of required memory.
- **Amount of stored data**: how much past data-item does the model need to retain?
- **Task boundaries**: does the model require a clear division among tasks?
- **Prediction oracle**: does the model require knowing the task identifier for prediction? If yes, the model is explicitly designed for Task-IL scenario.

## 2.5 State of the Art

A wide range of methods have been introduced in the last few years to address the problem of forgetting. These methods are typically classified

into three large groups, based on how they store or use task-specific information during the sequential learning process [18]. This section presents the most important CL methods for each category, with a brief description of them. We will encounter many of them in the following chapters, where they are compared against our methods. We report them also here to ease understanding for readers.

### 2.5.1 Regularization-based methods

The regularization-based methods aim to find a balance between retain the knowledge representation acquired from past task, while granting enough adaptive capacity to integrate new information into the model. This balance is most achievable if tasks exhibit substantial similarities in their complex feature embeddings, indicating that many acquired features can be mutually utilized across task.

Regularization techniques can be further divided into two subcategories:

- *Functional methods* are broadly inspired from knowledge distillation [19, 20]. They focus on maintaining consistent model outputs for previously seen task, storing the prediction of data samples and reusing them in the future as soft target using additional distillation losses.

- *Structural methods* emphasize rigid protection of model parameters. Generally, they identify a subset of network parameter highly correlated with the performance of each task at every level of the model architecture and prevent them from critical updates.

### Functional Methods

- **Learning without Forgetting (LwF)** [21] is probably the most famous functional method. Before learning the next task, the old model's predictions are stored and then reused during training as a form of pseudo-labels to distill prior knowledge. The aim of this type of regularization is to maintain the output related to previous tasks, thereby conserving earlier knowledge, even though these predictions (made before training on the current task) probably are no-sense.

- **Encoder Based Lifer Long Learning (EBLL)** [22] extends LwF using autoencoders to preserve knowledge from prior tasks when learning new ones. Each task has its own under-complete autoencoder to capture essential features. When faced with a new task, the feature reconstructions from these autoencoders are kept stable, ensuring preservation of crucial features of previous tasks.

### Structural methods

- **Elastic Weight Consolidation (EWC)** [23] preserves performance on previously learned task penalizing large changes to the neural network weights that are crucial for tasks the model has already learned. After the training of the current task, the importance of each weight for the task is computed using the Fisher Information Matrix. When the model is trained on the next task, EWC adds a regularization term to the loss function. This term penalizes changes to important weights, with the strength of the penalty determined by the previously computed importance values.

- **online Elastic Weight Consolidation (oEWC)** [24] is an efficient

approximation of the original EWC. In fact, EWC requires computing the Fisher Information Matrix for each task, a computation-intensive procedure whose cost is proportional to the number of tasks. This *online* variant of EWC approximates the original method with similar performance.

- **Synaptic Intelligence (SI)** [25] is conceptually similar to EWC, but SI operates in an online manner, trying to estimate the importance of the parameters during the training on the tastk itself.

### 2.5.2 Architectural methods

Architectural methods involve adapting or expanding the network's architecture to accommodate new tasks while preserving knowledge of previous tasks. These methods allow to dedicate distinct sets of parameters to every task. Upon encountering a new task, the model can create new submodules specifically for that task, allowing the network capacity to expand based on the number of the task to solve. While architectural approaches usually offer simple training procedures and are able to achieve high performance, they generally require the availability of the task-identifier at testing time, making them predominantly designed for the Task-IL scenario.

- **Progressive Neural Networks (PNN)** [26] were originally proposed to tackle Reinforcement Learning, but the method can be effectively adapted to other contexts, such as CL. When a new task arrives, instead of modifying the existing network, a new network (named *column*) specific for the new task is added. To employ the previous knowledge, the new network is linked to older columns by *lateral*

*connections*, that let the newer task to access information from the older tasks, but not vice versa. Although PNN is designed to prevent forgetting, it requires a significant amount of memory that increases linearly with the number of tasks.

- **Packing Multiple Tasks into a Single Network (PackNet)** [27] efficiently ”packs” knowledge of multiple tasks into a single neural network by pruning less important weights from earlier tasks and reusing this freed space for new tasks. It employs binary masks to determine which weights are preserved and which are adaptable. This approach avoids significant network expansion, making it memory-efficient while retaining prior knowledge.

- **Hard Attention to the Task (HAT)** [28] employs attention masks to define which parts of a neural network are active for a specific task. When training on a new task, the model learns both weights and these attention masks. The masks ensure only certain network regions are updated, preserving knowledge from previous tasks.

### 2.5.3 Replay-based methods

Replay-based methods counteract catastrophic forgetting by periodically replay some of the previously encountered data while learning new tasks. They are generally based on a small, fixed size buffer in which information from old data can be stored and reused in future. During the training of a new task, the model learns not only from the new data but also from random batches of old data fetched from this buffer. Despite lacking of correlation with biological insights, they generally outperform all regularization-based methods and, unlike architectural methods, do not re-

quire the information of task-ID at inference time. Their primary limitation is the need to maintain a memory buffer to store some samples, which could be raise privacy concerns in some real-world applications. To tackle this issue, a subcategory of methods known as *generative methods* opt for producing synthetic samples using GANs [29, 30], instead of storing real samples, as *rehearsal methods* do. However this approach introduces an additional level of complexity, as generative models trained in continual tend to collapse quickly. In fact, the results obtained by generative models are inferior to those of rehearsal methods generally.

- **Incremental Classifier and Representation Learning (iCaRL)** [31] utilizes a distillation loss term similar to LwF to prevent forgetting. It computes exemplars, representative samples from previous classes; when a new class is introduced, iCaRL updates features using new data and exemplars. Classification is based on a *nearest-mean-exemplars* classifier. To manage memory, it effectively selects the most representative exemplars, balancing memory constraints with knowledge retention.

- **Experience Replay (ER)** [32] is a pioneering work among the replay-based methods. It proposes to interleave training samples from the current task and past samples from the buffer in the training batches. The reservoirs sampling [33] can be adopted to randomly select images from the input stream, with the guarantee that all seen classes are equally represented in the buffer. ER has inspired most of the subsequent methods.

- **Gradient-based Sample Selection (GSS)** [34] introduces a specific optimization of the basic rehearsal formula meant to store maximally

informative samples in memory.

- **Gradient Episodic Memory (GEM)** [17] extends the use of a memory that gets replayed episodically with constraints on the gradients to be non-conflicting with updates for previous tasks.

- **Averaged Gradient Episodic Memory (A-GEM)** [35] proposes an efficient approximation of GEM, introducing significant improvements on computational and memory cost.

- **Function Distance Regularisation (FDR)** [36], similarly to LwF and iCaRL, introduces a distillation loss term in order to minimize interference of prior learning.

- **Hindsight Anchor Learning (HAL)** [37] individuates synthetic replay data points that are maximally affected by forgetting.

- **Dark Experience Replay (DER)** [38], and its improved versions **DER++** [38] and **X-DER** [39] are enhanced versions of ER. Along with the samples in the buffer, these methods also store previous network responses and with additional self-distillation loss terms, they push the model to replicate the same outputs.

- **Experience Replay with Asymmetric Cross-Entropy (ER-ACE)** [40] is an enhanced version of ER that proposes to separate the contributions of cross-entropy loss of samples in the buffer from those of the input stream, aimed at preventing imbalances due to the simultaneous optimization of current and past data.

- **Contrastive Continual Learning (CO$^2$L)** [41] proposes to facilitate knowledge transfer from samples stored in the buffer by opti-

mizing a contrastive learning objective [42], avoiding any potential bias introduced by a cross-entropy objective. To perform classification, a linear classifier needs to be first trained on the samples stored in the buffer.

- **DualNet** [43] is a dual-backbone architecture decoupling the issue of incremental classification from the one of learning an overall transferable representation. The latter task is demanded to one of the backbones (*slow learner*), trained with a self-supervised loss term on i.i.d. data coming from the replay buffer; the other backbone (*fast learner*) is instead tasked with fitting the CL tasks while taking advantage of the representations produced by the slow learner.

- **Continual Prototype Evolution (CoPE)** [44] proposes a classifier based on class prototypes, whose careful update scheme allows for learning incrementally while avoiding sudden disruptions in the latent space.

# Part II

# EXPLOITING PRIOR/ADDITIONAL EXPERIENCES FOR CONTINUAL LEARNING

*"I have to believe in a world outside my own mind. I have to believe that my actions still have meaning, even if I can't remember them. I have to believe that when my eyes are closed, the world's still there. Do I believe the world's still there? Is it still out there? ... Yeah. We all need mirrors to remind ourselves who we are. I'm no different ... now ... where was I?"*

Leonard "Lenny" Shelby, *Memento*, 2000

A fundamental truth of human cognition is that no one starts as a *blank slate*. Our lives are complex and full of innate knowledge and instinctual understanding from the beginning. And this foundational knowledge is not static. It keeps evolving, adapting and growing, drawing from each single experience, each challenge and each learned lesson. It is reservoir of accumulated knowledge that enables people to undertake new tasks, find innovative solutions to unprecedent challenges, and continue to broaden the bounds of our knowledge.

This inherent human ability of using past experiences as a foundation for future learning serves as our inspiration for the following Chapters.

In Chapter 3 we highlight the importance of prior knowledge – specifically, how past experience can be used to enable more efficient and effective problem solving. Therefore, we emulate the reactivation of prior knowledge using an auxiliary data stream.

In Chapter 4 while we strive with emulating human cognition by leaning on pre-trained models, we address the inherent limitations of Transfer Leaning in the CL paradigm. The challenge lies in observing that the impact of pre-training decreases as the number of tasks increases. We propose a novel method where each task of the sequence can equally leverage prior knowledge from pre-training.

Chapter 5 presents a shift in perspective, where we face CL by pairing the classification task with an auxiliary task, from which some knowledge can be re-used to address the primary task. This synergy is designed to guide the learning process, with particular emphasis on the use of self-supervised equivariant tasks.

CHAPTER

# THREE

# LEVERAGING PAST KNOWLEDGE THROUGH AUXILIARY DATA

Our journey in tackling catastrophic forgetting starts with a work that aims at includes principles of human learning, specifically trying to harness prior knowledge derived from past experiences that can be reused for learning new tasks and solving new problems more effectively. In our first attempt, the past knowledge is emulated through the use of an auxiliary data stream. By incorporating such auxiliary information during training, the model becomes more adept at disentangling underlying patterns and generalizing knowledge. As a result, when new tasks arise, the model can leverage its enriched understanding of shared features to adapt and learn more efficiently.

## 3.1 Motivation

Human beings and animals are naturally able to memorize information presented in a sequence [45]; on the contrary, Artificial Neural Networks (ANNs) learning from a non-i.i.d. stream of data incur in *Catastrophic Forgetting* [5, 6]. Continual Learning (CL) [46, 44] aims at designing methods that compensate for this issue and facilitate the retention of previous knowledge either by means of regularization [23, 21], architectural designs [26, 28] or (pseudo-)replay of past data [6, 47, 29].

The insurgence of catastrophic forgetting is ascribed to the tendency of models to rewrite their hidden representations as they adjust their parameters to best fit an input distribution that changes in time [48]. However, *McRae & Hetherington* highlight a meaningful difference in the way humans and ML models learn from a sequence of data: whenever human subjects are evaluated on their ability to memorize a sequence of concepts, they start out possessing an already-large body of knowledge [49]. In other words, humans are generalists that can anchor novel data in the context of previous knowledge, while ANNs must specialize on a limited pool of data at each time without any additional reference.

An obvious choice to bridge this gap is *pre-training* the models on a large amount of available off-the-shelf i.i.d. data, leading to a better initialization for the learning procedure [49, 50]. However, we observe that pre-training is not always rewarding in a CL setting, especially in case of small-size replay memories: the ever-changing stream of data entails large changes in model parameters, leading to the forgetting of the pre-training.

We instead propose a learning strategy to limit catastrophic forgetting by providing an additional data stream (uncorrelated from the target data), from which the network can draw auxiliary knowledge. The role of this

data stream is to provide models with a more stable representation of the world that can be re-used for incrementally learning new classes or categories leveraging the already-learned low-level features. Indeed, it appears that the human brain can adapt and rewire itself more easily when learning new things related to familiar skills because pre-existing neuronal structure constrains what one can learn [51]. We attempt to enforce this concept into CL through the definition of *an associative rule* that helps learning new classes by measuring the simultaneous firing of neurons between past knowledge and the current data stream. This is implemented through a simple yet effective strategy named **MAH**, that, during a new task, assigns new classes to model's corresponding **M**ost **A**ctivated **H**eads.

Experimental results carried out on standard CL settings, involving CIFAR-10 and (a subset of) Tiny-ImageNet benchmarks, demonstrate that using a separate auxiliary data stream is mostly beneficial with limited size buffer leading to a performance gain of several percent points w.r.t. state-of-the-art methods. Analogously, the MAH strategy reveals to be more effective than the standard class mapping procedure independently from the buffer size. We also investigate the role of model pre-training as compared to sustained auxiliary data employment highlighting that, for small-size buffers, auxiliary data is to be preferred to pre-training.

Our strategy is beneficial in Continual Learning from multiple perspectives: the model avoids overfitting current examples, learns more general features and – as auxiliary data-points stand in for future examples – better prepares to learn future classes by suitably associating past knowledge to the new acquired one. All these aspects are mainly observed with reduced buffer size, thus contributing to the efforts that aim at generalizing CL approaches to real-world scenarios.

Our contribution can be summarized as follows:

- We propose **Most Activated Heads** (MAH) strategy, that aims at mitigating the problem of Catastrophic Forgetting by using an auxiliary stream of data during training and by effectively assigning each new class of the main stream to the most appropriate model classification head.

- Compared to other rehearsal-based methods, our MAH achieves state-of-the-art results on several continual learning benchmarks. Significantly, our MAH scores remarkably results when used with small buffer, suggesting it is efficient in retaining and utilizing past information with limited memory capacity.

- We conduct a detailed analysis of the benefits of using auxiliary data over a pre-trained network and find that for limited-size buffer using an external stream leads to better results. In addition, auxiliary data can be synthesized through generative models maintaining comparable results, freeing us from the need to store auxiliary data for the training period.

## 3.2 Related Work

The seminal study by *McCloskey and Cohen* first drew attention to the tendency of ANNs to forget previously learned knowledge catastrophically [5]. In spite of the outstanding results achieved by deep learning models in recent years [52, 53], this problem still persists and prevents ANNs from learning flexibly from non-i.i.d. data-streams. To tackle this issue, researchers and practitioners design CL methods, i.e., strategies that make machine learning models retain high accuracy on previously seen

data when trained on an ever-changing input distribution [46, 44]. While many distinct strategies have been applied for this purpose, CL approaches can be broadly categorized into two families: *structuring* or *generalization*.

### 3.2.1 Structuring approaches to Continual Learning

Methods in the first class aim at making interference between distinct concepts less likely by endowing the stored knowledge with a disentangled structure. [54] first pioneered the idea to reduce forgetting by orthogonalizing feature representations. A similar approach was recently taken by [55]. Alternatively, *structuring* can be pursued at an architectural level, by explicitly allocating distinct subsets of model parameters to distinct tasks [26, 27, 56], encouraging non-overlapping activation patterns for different data [28, 57], or by simply applying dropout [58, 59]. Finally, several approaches regularize back-propagation by projecting the gradient to minimize the interference between tasks learned at different times [17, 60, 61]. While *structuring* approaches are usually characterized by a simpler training procedure, they typically require the availability of a task-identifier at test time.

### 3.2.2 Generalization approaches to Continual Learning

At the opposite end of the spectrum, *generalization* methods prevent forgetting by encouraging the model to compare and contrast input data encountered throughout the sequence, thus recovering the i.i.d. property of training [49]. Most notably, rehearsal-based approaches do so by maintaining a working memory of previously seen examples and interleaving them with the input data [6, 47, 38, 34, 62], while pseudo-rehearsal meth-

ods approximate this procedure with a generative model [63, 29]. Other works prioritize the learning of high-level representations either by adopting learning objectives designed not to disrupt the performance on previous tasks [31, 64, 21, 65], or by making use of semi-supervised learning techniques to learn general features [41, 43, 66]. The *generalization* approach is taken to an extreme by [67], which shows satisfactory results on CL benchmarks by training a model in an i.i.d. fashion on samples gathered greedily from the input stream.

*Generalization* strategies naturally blend knowledge gathered at different times to build a unified predictor, making them more reliable than their *structuring* counterparts in the realistic settings where no task-identifier is given at testing-time [68, 8]. The approach proposed in this paper aligns with the former group of methods; indeed, we argue that generalization should be extended beyond already-seen data and embrace yet-unseen knowledge as well.

## 3.3 Method

Most CL methods use current and, if the method implies a rehearsal strategy, past task classification heads during the training of the current task. Future heads, that will be mapped to classes from following tasks, are not involved in the process at all. This poses a potentially dangerous situation due to the model minimizing its prediction scores for future heads, which results in a high loss peak when these heads are used at the beginning of future tasks.

We propose to leverage an auxiliary data stream, not correlated with the main task stream, in order to keep these future heads activated since

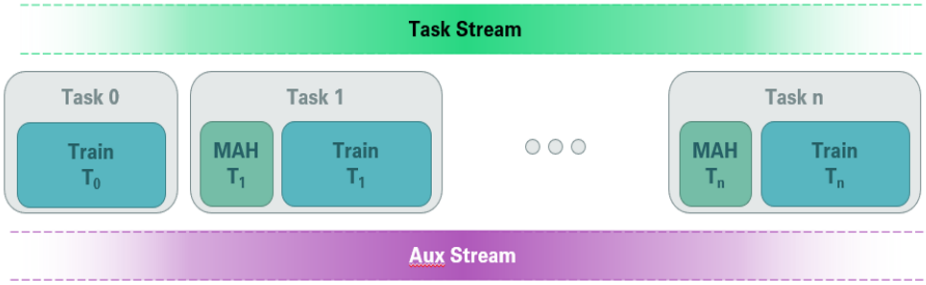



***Figure 3.1:*** *We propose to mitigate forgetting by introducing an extra data stream, which serves as source of auxiliary knowledge for the network. This auxiliary stream plays a crucial role in offering the network a more consistent representation of the world, which can be utilize to learn new classes sequentially while leveraging the previous acquired low-level features.*

the beginning of training. The proposed strategy is also beneficial to learn more distinguishing and reusable features, as the model cannot focus on simply discriminating between the classes from the task at hand. Furthermore, since auxiliary training leads future task heads to learn to recognize their own specific patterns, we exploit this property to devise a "most activated heads" (MAH) assignment strategy for future classes, that minimizes the loss peak that the model typically incurs at the beginning of a new task. Hence, the use of an auxiliary stream favors the current task and improves forward transfer to future tasks. The proposed approach is illustrated in Fig. 3.2.

Formally, a typical CL classification problem requires solving several tasks sequentially, where each task $T_t$, with $t \in \{1, ..., T\}$ and $T$ being the number of tasks, consists in learning to classify a set of classes $\mathbf{C}_t$. In

this work, we follow the common Class-IL and Task-IL settings [8], which assume no overlap between classes from different tasks.

Each task is associated with an i.i.d. distribution $D_t$ of $(\mathbf{x}, y)$ pairs of a data point with the corresponding class label from $\mathbf{C}_t$. In practice, the distribution is approximated by a finite set of samples, i.e., $\mathbf{D_t} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_{N_t}, y_{N_t})\}$, where $N_t$ is the number of examples for task $t$.

The objective of CL is to find a function $f_\theta$, depending on a set of learnable parameters $\theta$, that minimizes a classification objective over the entire task sequence, such as:

$$\arg\min_{\theta} \sum_{t=1}^{T} \sum_{(\mathbf{x}_i, y_i) \in \mathbf{D_t}} \mathcal{L}_{\mathrm{C}} \left( f_\theta(\mathbf{x}_i), y_i \right), \tag{3.1}$$

where $\mathcal{L}_{\mathrm{C}}$ is the classification loss (e.g., cross-entropy).

While training for the current task, most recent CL approaches [23, 21, 47, 38] attempt to reduce forgetting by adding an additional loss term that attempts to retain accuracy on previously-seen tasks. The in-task objective at task $t$ then becomes:

$$\arg\min_{\theta} \sum_{(\mathbf{x}_i, y_i) \in \mathbf{D_t}} \mathcal{L}_{\mathrm{C}} \left( f_\theta(\mathbf{x}_i), y_i \right) + \mathcal{L}_{\mathrm{CL}}, \tag{3.2}$$

where $L_{CL}$ is a generic additional loss term that implements countermeasures against catastrophic forgetting and may vary depending on the specific method. For example, in rehearsal-based approaches, $L_{CL}$ could be an additional cross-entropy loss term computed on buffered samples from previous task, or it could be a distillation loss that aims to match current network's outputs with past ones on the same samples, as used in [38].

In the proposed scenario, an additional distribution of i.i.d. auxiliary data $A$, where $A \neq D_t \ \forall t$, is available to the model at training

time. Again, the distribution is represented by a set of sample/label pairs $\mathbf{A} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_{N_t}, y_{N_t})\}$, where labels belong to the class set $\mathbf{C}_A$.

In the following, we explain the two key aspects of the proposed approach: head pre-activation and "most activated heads" class mapping.

### 3.3.1 Head pre-activation

To ensure that the model employs all of its classification heads from the start, we use classes from the auxiliary dataset as "place-holders" for classes from future tasks.

The basic requirement of an auxiliary dataset $\mathbf{A}$ is related to the cardinality of its set of classes, $|\mathbf{C}_A|$, which should satisfy the following condition:

$$|\mathbf{C}_A| \geq \sum_{t=2}^{T} |\mathbf{C}_t| . \tag{3.3}$$

In other words, the number of auxiliary classes should be at least equal to the total number of classes in the sequence of continual learning tasks, minus the number of classes from the first task. This guarantees that, when training on the first task, the auxiliary dataset provides enough classes for the classification heads reserved to future tasks.

*Before* starting to train on the first task $t = 1$, we randomly choose a subset $\mathbf{C}_{A,t} \subseteq \mathbf{C}_A$, with cardinality $|\mathbf{C}_{A,t}| = \sum_{t=2}^{T} |\mathbf{C}_t|$. Samples from the selected classes are included in the auxiliary sub-dataset $\mathbf{A}_t$ and class indexes from $\mathbf{C}_{A,t}$ are re-mapped to the indexes of classes in $\cup_{t=2}^{T} \mathbf{C}_t$ corresponding to future tasks.

At task $t$, we merge the corresponding dataset $\mathbf{D}_t$ and the auxiliary sub-dataset $\mathbf{A}_t$ and train on the joint set of classes, in order to minimize

the following new in-task objective:

$$\arg\min_{\theta} \sum_{(\mathbf{x}_i, y_i) \in \mathbf{D}_t \cup \mathbf{A}_t} \mathcal{L}_{\mathrm{C}} \left( f_\theta(\mathbf{x}_i), y_i \right) + \mathcal{L}_{\mathrm{CL}}. \tag{3.4}$$

As a result, we ensure that all classification heads are employed, reducing the risk of loss peaks on new tasks, and encourage the model to learn more complex, discriminative and stable features.

### 3.3.2 "Most activated heads" (MAH) class mapping

At the beginning of each task $t > 1$, it is necessary to update the set of auxiliary classes in $\mathbf{A}_t$, since $|\mathbf{C_t}|$ classes must be removed to make room for classes from the new task.

Moreover, in this scenario, it also makes sense to *assign* the specific heads that will correspond to classes in the new task, rather than simply associating them to the next available heads. An appropriate class mapping can make better (re)use of features learned by the model for classification of auxiliary classes, and reduce high losses that may lead to forgetting previously-learned features.

Our head assignment approach, named MAH from "most activated heads", acts before beginning to train on task $t > 1$, by first computing the average logits $\mathbf{l}_c$, i.e., pre-softmax head activations, for each task class $c \in \mathbf{C_t}$: to this aim, we select the subset $\mathbf{D}_{t,c} \subset \mathbf{D_t}$ which only contains elements of class $c$, and average the corresponding logit vectors as returned by model $f_\theta$:

$$\mathbf{l}_c = \frac{1}{N_{t,c}} \sum_{(\mathbf{x}_i, y_i) \in \mathbf{D}_{t,c}} f_\theta \left( \mathbf{x}_i \right), \tag{3.5}$$

where $N_{t,c}$ is the number of elements of class $c$ in $\mathbf{D_t}$.

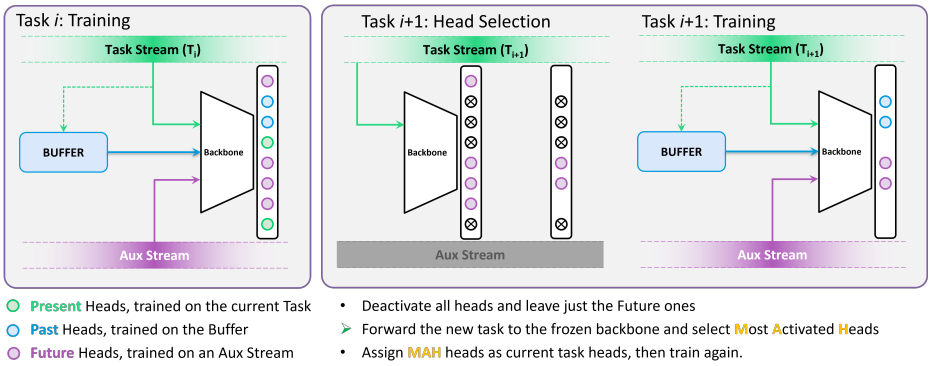



***Figure 3.2:*** *During the training of the current Task $T_i$ the model is trained combining the "current" data coming from the Task Stream and the "past" data stored inside the buffer. In addition, the remaining heads are trained using the auxiliary data stream. At the beginning of a new Task $T_{i+1}$ the MAH procedure is conducted as follows: 1) Only the heads trained with the auxiliary data are kept activated; 2) the $T_{i+1}$ task is forwarded to the frozen model in order to store activation information about the heads; 3) for each class in task $T_{i+1}$, each new class is assigned to the head that activates the most, replacing the corresponding auxiliary data class.*

Then, each new class in $\mathbf{C_t}$ is simply associated to the classification head that maximizes its predicted score, i.e., $\arg\max \mathbf{l}_c$. In case of index collisions, largest values are given priority. Finally, the new auxiliary sub-dataset $\mathbf{A}_t$ is updated by removing from $\mathbf{A}_{t-1}$ the set of classes corresponding to selected indexes, i.e., $\{\arg\max \mathbf{l}_c\}_{c \in \mathbf{C_t}}$, and training proceeds as previously described.

## 3.4 Experimental Result

### 3.4.1 Datasets

We focus our experiments on two common evaluation protocols [8]: class-incremental (Class-IL), where the model is asked to gradually solve the complete problem but classes become available at different times; task-incremental (Task-IL), where the model is guided by the task-identity and can only focus to solve each task independently. Specifically, we leverage **Seq-CIFAR-10** [25], a widely-used image classification dataset obtained by splitting the 32×32 images of CIFAR-10 into 5 binary tasks. For a more comprehensive evaluation, we also test on the larger 64×64 **Seq-Micro-ImageNet**: a novel benchmark composed of a 20-class subset of Tiny-ImageNet [15], split into 5 tasks of 4 classes each.

As for the choice of the auxiliary data, we pair the original data with similarly-sized datasets. In particular, the auxiliary dataset for Seq-CIFAR-10 consists of a subset of 10 super-classes from CIFAR-100, selected among those which are not semantic-related to those contained in CIFAR-10. For Seq-Micro-ImageNet, we select a subset of 20 classes from ILSVRC-2012, making sure that the chosen data is as unrelated as possible with the original Tiny-ImageNet classes. In detail, we first remove Tiny-ImageNet classes from the entire label set; then, we group the remaining 800 classes into 293 super-classes, corresponding to synsets found at distance 8 from the *entity* root node. Finally, we apply Spectral Clustering to select the 20 classes which are most representative of the super-classes.

### 3.4.2 Training procedure

We apply the approach described in Section 3.3 by adapting the DER++ [38] method, a recent rehearsal-based approach inspired by knowledge distillation principles. For a fair comparison among different models, in our experiments we follow [38] and adopt the same training settings. As backbone, we use ResNet-18 [69] (not pre-trained). We optimize our model with SGD, for 50 epochs on Seq-CIFAR-10 and 100 on Seq-Micro-ImageNet. During training, samples from the current task and from auxiliary classes are combined so that each mini-batch contains data from both domains. We apply random crops and horizontal flips as data augmentation. All hyperparameters are as defined in [38].

### 3.4.3 Results

To validate the effectiveness of our approach using auxiliary data during training, we compare our method with other CL methods based on rehearsal strategies: ER [70], GEM [17], A-GEM [60], iCaRL [31], FDR [36], GSS [34], HAL [37], DER [38] and vanilla DER++ [38]. Performance for these methods is reported from [38], except for the setup with buffer size equal to 50[1].

As performance metrics, we report the *Final Average Accuracy* in the Class-IL and in the Task-IL settings.

Table 3.1 and Table 3.2 report results on Seq-CIFAR-10 and Seq-Micro-ImageNet, respectively. Our method yields the best Class-IL performance when tested with small/medium buffer size. It is also noteworthy

[1] In this case, results were computed using the Mammoth framework for PyTorch: https://github.com/aimagelab/mammoth

| *Class-IL* | | | | |
|---|---|---|---|---|
| Buffer size | 50 | 200 | 500 | 5120 |
| ER [70] | 32.69 ± 0.39 | 44.79 ± 1.86 | 57.74 ± 0.27 | 82.47 ± 0.52 |
| GEM [17] | 22.10 ± 0.41 | 25.54 ± 0.76 | 26.20 ± 1.26 | 25.56 ± 3.46 |
| A-GEM [60] | 20.02 ± 0.08 | 20.04 ± 0.34 | 22.67 ± 0.57 | 21.99 ± 2.29 |
| iCaRL [31] | *55.51 ± 1.64* | 49.02 ± 3.20 | 47.55 ± 3.95 | 55.07 ± 1.55 |
| FDR [36] | 28.32 ± 4.51 | 30.91 ± 2.74 | 28.71 ± 3.23 | 19.70 ± 0.07 |
| GSS [34] | 26.62 ± 1.36 | 39.07 ± 5.59 | 49.73 ± 4.78 | 67.27 ± 4.27 |
| HAL [37] | 25.26 ± 1.73 | 32.36 ± 2.70 | 41.79 ± 4.46 | 59.12 ± 4.41 |
| DER [38] | 44.85 ± 2.71 | 61.93 ± 1.79 | 70.51 ± 1.67 | 83.81 ± 0.33 |
| DER++ [38] | 49.28 ± 3.16 | *64.88 ± 1.17* | *72.70 ± 1.36* | **85.24 ± 0.49** |
| **Ours** | **56.33 ± 0.95** | **70.86 ± 0.95** | **75.07 ± 0.41** | *84.56 ± 0.55* |
| *Task-IL* | | | | |
| Buffer size | 50 | 200 | 500 | 5120 |
| ER [70] | 86.98 ± 1.19 | 91.19 ± 0.94 | 93.61 ± 0.27 | **96.98 ± 0.17** |
| GEM [17] | 81.36 ± 1.43 | 90.44 ± 0.94 | 92.16 ± 0.69 | 95.55 ± 0.02 |
| A-GEM [60] | 81.09 ± 1.88 | 83.88 ± 1.49 | 89.48 ± 1.45 | 90.10 ± 2.09 |
| iCaRL [31] | *88.86* ± 2.51 | 88.99 ± 2.13 | 88.22 ± 2.62 | 92.23 ± 0.84 |
| FDR [36] | 85.23 ± 1.24 | 91.01 ± 0.68 | 93.29 ± 0.59 | 94.32 ± 0.97 |
| GSS [34] | 85.22 ± 1.03 | 88.80 ± 2.89 | 91.02 ± 1.57 | 94.19 ± 1.15 |
| HAL [37] | 78.73 ± 3.16 | 82.51 ± 3.20 | 84.54 ± 2.36 | 88.51 ± 3.32 |
| DER [38] | 85.04 ± 1.17 | 91.40 ± 0.92 | 93.40 ± 0.39 | 95.43 ± 0.33 |
| DER++ [38] | 86.14 ± 2.56 | *91.92 ± 0.60* | **93.88 ± 0.50** | *96.12 ± 0.21* |
| **Ours** | **89.57 ± 2.47** | **93.30 ± 0.64** | *93.62 ± 0.58* | 95.84 ± 0.42 |

***Table 3.1:*** *Final Average Accuracy (FAA) [↑] on Seq-CIFAR-10 for several replay-based Continual Learning methods. Best results in bold, second-best in italic.*

that as the performance gain of our approach increases as the size of the buffer decreases. When buffer size becomes significantly larger, vanilla DER++ still achieves the best results, showing that retaining and replaying enough data (5,120 samples represent more than 10% of the entire training set of CIFAR-10) is still the best option to alleviate catastrophic forgetting, although this goes in stark contrast to generalizing continual learning methods to real-world problems. A similar behavior can be observed on the simpler Task-IL setting, where our method obtains the highest performance or is on par with existing methods under low–data availability regimes.

We also compare our approach to Co$^2$L [41], that recently achieved state-the-art performance in both settings[2]. Nevertheless, on Seq-CIFAR-10, our method yields better Class-IL performance than Co$^2$L [41], that reaches 65.57$\pm$1.37 with buffer size of 200 and 74.26$\pm$0.77 with buffer size of 500, respectively compared 70.86$\pm$0.95 and 75.07$\pm$0.41 by our method. The lower standard deviation also shows that our approach tends to be more stable across tasks.

Finally, we monitor the loss over consecutive tasks in order to evaluate the impact of auxiliary data during training. Fig. 3.3 shows the average training loss for vanilla DER++ and our method on Seq-Micro-ImageNet. It can be observed that, as new tasks come in (every 100 epochs), the proposed approach shows a smoother loss surface, conversely to the vanilla counterpart that, instead, exhibits more noticeable peaks. Thus, the proposed strategy also improves forward transfer and prevents disrupting gradient peaks when the model switches to new tasks, resembling non-continual learning scenarios.

[2] Co$^2$L is not reported in Tables 3.1 and 3.2, as its training strategies are significantly different from the methods shown in those tables.

| Buffer | Method | *Class-IL* | *Task-IL* |
|---|---|---|---|
| 50 | ER [70] | 22.58 ± 0.71 | 66.28 ± 1.83 |
| | DER [38] | 29.35 ± 2.16 | 70.88 ± 0.83 |
| | DER++ [38] | 31.92 ± 2.15 | 69.86 ± 1.96 |
| | **Ours** | **37.24 ± 2.76** | **71.84 ± 1.82** |
| 50 | ER [70] | 36.22 ± 1.06 | 77.82 ± 1.24 |
| | DER [38] | 46.18 ± 1.44 | 81.12 ± 1.59 |
| | DER++ [38] | 51.84 ± 1.32 | **82.66 ± 1.60** |
| | **Ours** | **52.83 ± 0.83** | 78.48 ± 1.42 |
| 50 | ER [70] | 49.70 ± 0.71 | 84.35 ± 0.79 |
| | DER [38] | 56.58 ± 2.44 | 84.56 ± 1.19 |
| | DER++ [38] | **60.28 ± 2.31** | **85.10 ± 0.93** |
| | **Ours** | 59.36 ± 0.81 | 79.95 ± 0.61 |
| 50 | ER [70] | 70.40 ± 1.30 | 89.20 ± 0.01 |
| | DER [38] | 68.75 ± 0.25 | 89.35 ± 0.85 |
| | DER++ [38] | **74.98 ± 0.66** | **90.72 ± 0.65** |
| | **Ours** | 71.65 ± 0.82 | 85.93 ± 1.29 |

***Table 3.2:*** *Final Average Accuracy (FAA) [↑] on Seq-Micro-ImageNet for Experience Replay-based methods.*

### 3.4.4 Ablation study

In order to substantiate our design choices, we perform an ablation study to quantify the contribution of a) using the auxiliary data stream and b) the MAH strategy. The obtained results are reported in Table 3.3 and compared

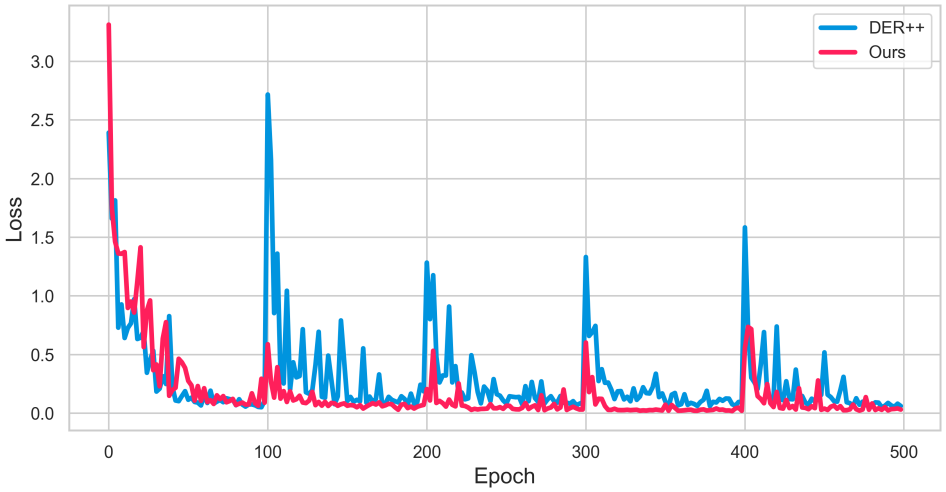



***Figure 3.3:*** *Training loss trend for our approach (red) and DER++ (blue) on Seq-Micro-ImageNet. When the model switches to a new task, MAH reduces loss peaks by assigning new classes to the most suitable available heads trained on auxiliary data.*

to vanilla DER++ and DER++ with auxiliary data but **without MAH**. When the MAH strategy is not used, the classification heads are selected in a sequential order without making use of neural activation mapping between past (auxiliary) and current classes. Results show that training with auxiliary data yields significant performance gains for all buffer sizes but 5,120, where replayed knowledge becomes prevalent. In all cases, MAH outperforms sequential head mapping.

### 3.4.5 Effect of Pre-training

We further investigate whether it is better to employ a backbone pre-trained on auxiliary data or to train it from scratch using the proposed strategy. The

| Buffer | Method | *Class-IL* | *Task-IL* |
|---|---|---|---|
| 50 | DER++ | 49.28 ± 3.16 | 86.14 ± 2.56 |
| | ↪ + AUX | 52.74 ± 1.02 | 88.51 ± 2.01 |
| | ↪ + MAH | **56.33 ± 0.95** | **89.57 ± 2.47** |
| 200 | DER++ | 64.88 ± 1.17 | 91.92 ± 0.60 |
| | ↪ + AUX | 69.91 ± 1.48 | 92.57 ± 1.02 |
| | ↪ + MAH | **70.86 ± 0.95** | **93.30 ± 0.64** |
| 500 | DER++ | 72.70 ± 1.36 | 93.88 ± 0.50 |
| | ↪ + AUX | 74.24 ± 0.61 | **93.93 ± 0.47** |
| | ↪ + MAH | **75.07 ± 0.41** | 93.62 ± 0.58 |
| 500 | DER++ | **85.24 ± 0.49** | **96.12 ± 0.21** |
| | ↪ + AUX | 84.26 ± 0.22 | 95.58 ± 0.22 |
| | ↪ + MAH | 84.56 ± 0.55 | 95.84 ± 0.42 |

***Table 3.3:*** *Ablation Study. Final Average Accuracy (FAA) [↑] obtained by the vanilla DER++ (first row of each block), DER++ with auxiliary data (second row), and the proposed method (third row), combining DER++ with auxiliary data and MAH strategy, for different buffer sizes.*

results of this analysis are reported in Table 3.4: pre-training on auxiliary data appears to be always beneficial compared to training from scratch in DER++ and as the buffer size increases. On the contrary, on small buffers, using auxiliary data with our approach on a model trained from scratch yields better performance than pre-training. Furthermore, with our method, pre-training on auxiliary data leads instead to lower performance than training from scratch, showing that pre-training is not always a reliable alternative to continuously training with auxiliary data.

### 3.4.6 Generative Auxiliary Model

In the previous sections, we have consistently observed that using auxiliary data helps retaining knowledge of previous tasks, especially with limited buffer size. However, it is not efficient to maintain the auxiliary data in memory, as in that case it is still preferable to simply use a larger buffer.

A viable alternative would be to replace the auxiliary stream with a generative replay model and use generated samples during task training. In order to investigate the feasibility of this option, we use a generative adversarial network (GAN) [71] to learn the distribution of auxiliary data, and employ synthetic images in place of real ones in Eq. 3.4 in our method.

We then replicate the experiments carried out on Seq-CIFAR-10 by pre-training a BigGAN model [72] on super-classes of CIFAR-100 used as auxiliary data, and compare the results with those obtained when using real images. As it can be seen in Table 3.5, performance achieved when using generated images follow the same behavior observed with real data, i.e., performance increase in settings with small buffer size, while the approach is less beneficial when the replay memory increases.

| Buffer | Method | *Class-IL* | *Task-IL* |
|---|---|---|---|
| 50 | DER++ | 49.28 ± 3.16 | 86.14 ± 2.56 |
| | DER++ with pre-training | 50.82 ± 3.34 | 84.02 ± 2.98 |
| | Ours | **56.33 ± 0.95** | **89.57 ± 2.47** |
| | Ours with pre-training | 53.52 ± 3.60 | 89.52 ± 0.88 |
| 200 | DER++ | 64.88 ± 1.17 | 91.92 ± 0.60 |
| | DER++ with pre-training | 68.71 ± 1.01 | 92.43 ± 0.53 |
| | Ours | **70.86 ± 0.95** | **93.30 ± 0.64** |
| | Ours with pre-training | 65.17 ± 2.67 | 91.35 ± 1.71 |
| 500 | DER++ | 72.70 ± 1.36 | 93.88 ± 0.50 |
| | DER++ with pre-training | **75.91 ± 0.26** | **94.39 ± 0.29** |
| | Ours | 75.07 ± 0.41 | 93.62 ± 0.58 |
| | Ours with pre-training | 71.39 ± 2.77 | 91.77 ± 0.76 |
| 5120 | DER++ | 85.24 ± 0.49 | 96.12 ± 0.21 |
| | DER++ with pre-training | **86.60 ± 0.42** | **96.29 ± 0.09** |
| | Ours | 84.56 ± 0.55 | 95.84 ± 0.42 |
| | Ours with pre-training | 82.20 ± 1.37 | 94.65 ± 0.36 |

***Table 3.4:*** *Effect of Pre-training. Final Average Accuracy (FAA) [↑] obtained by the vanilla DER++ (first row of each block), DER++ pre-trained with auxiliary data (second row), the proposed method trained from scratch (third row), and the proposed model pre-trained with auxiliary data (fourth row), for different buffer sizes.*

| Buffer | Aux. data | *Class-IL* | *Task-IL* |
|---|---|---|---|
| 50 | *none* | 49.28 ± 3.16 | 86.14 ± 2.56 |
| | real | 56.33 ± 0.95 | 89.57 ± 2.47 |
| | synthetic | 54.47 ± 3.05 | 89.23 ± 1.83 |
| 200 | *none* | 64.88 ± 1.17 | 91.92 ± 0.60 |
| | real | 70.86 ± 0.95 | 93.30 ± 0.64 |
| | synthetic | 68.84 ± 0.77 | 92.85 ± 0.24 |
| 500 | *none* | 72.70 ± 1.36 | 93.88 ± 0.50 |
| | real | 75.07 ± 0.41 | 93.62 ± 0.58 |
| | synthetic | 74.35 ± 1.04 | 93.42 ± 0.40 |
| 5120 | *none* | 85.24 ± 0.49 | 96.12 ± 0.21 |
| | real | 84.56 ± 0.55 | 95.84 ± 0.42 |
| | synthetic | 84.14 ± 0.45 | 95.65 ± 0.23 |

***Table 3.5:*** *Effect of replacing auxiliary data with a GAN. Final Average Accuracy (FAA) [↑] obtained on Seq-CIFAR-10 when using no auxiliary data (vanilla DER++), real auxiliary data (the proposed method) and synthetic data.*

## 3.5 Discussion

In this work, we propose a novel approach for improving continual learning accuracy by leveraging external data. Our experiments show that providing the model with an additional auxiliary stream leads to an increase in performance, especially when the employed memory buffer is small, and to more stable training at the beginning of each task. We also observe that our approach works consistently better, on small buffer settings, than alternative knowledge transfer strategies such as direct pre-training on auxiliary data. The approximation of the auxiliary data distribution through the use of generative models also outperforms state-of-the-art models, thus indicating the future direction of this work, i.e., a more effective modeling of previous real-world knowledge as it seems to happen in the human hippocampus [73].

## 3.6 Publications

Bellitto, G., Pennisi, M., Palazzo, S., Bonicelli, L., Boschini, M., Calderara, S., & Spampinato, C. (2022). Effects of auxiliary knowledge on continual learning. In 2022 26th International Conference on Pattern Recognition (ICPR) (pp. 1357-1363). IEEE.

CHAPTER

# FOUR

# LEVERAGING PAST KNOWLEDGE THROUGH PRE-TRAINING

In the previous chapter, we shown how the use of past knowledge can be emulated through the use of auxiliary data. Another alternative solution is given by transferring and reusing knowledge from models trained on different data domains, as typically done in Transfer Learning. In this case, the simplest approach is to pre-train a neural network on a *source* domain, and then fine-tune the same on a *target* domain. Unfortunately, applying this type of approach straightforwardly in Continual Learning, turns out to be not as effective as it is for other tasks. The main reason is that even pre-training incurs in forgetting: only the first task can take fully advantage of pre-training; for subsequent tasks, the effect of pre-training becomes increasingly marginal.

In this chapter we investigate the entanglement between Continual Learning and Transfer Learning. We propose a new method, specifically

designed for a Continual Learning scenario, with the aim that the entire tasks sequence, equally for the first task as well as for the subsequent ones, can fully exploit past knowledge deriving from pre-training.

## 4.1 Motivation

Thanks to the enthusiastic development carried out by the scientific community, there exist a myriad of deep learning models that can be either readily deployed or easily adapted to perform complex tasks [74, 52, 53, 75, 76]. However, the desiderata in practical applications [77] often oversteps the boundaries of the typical *i.i.d.* paradigm, fostering the study of different learning approaches.

In contrast with the natural tendency of biological intelligence to seamlessly acquire new skills and notions, as we have already widely discussed, deep models are prone to *catastrophic forgetting* [5], *i.e.*, they fit the current input data distribution to the detriment of previously acquired knowledge. In light of this limitation, the sub-field of Continual Learning (CL) [18, 46, 8] aspires to train models capable of adaptation and lifelong learning when facing a sequence of changing tasks, either through appositely designed architectures [26, 24, 27], targeted regularization [21, 23, 25] or by storing and replaying previous data points [31, 70, 38, 47].

On a similar note, human intelligence is especially versatile in that it excels in contrasting and incorporating knowledge coming from multiple domains. Instead, the application of deep supervised learning algorithms typically demands for large annotated datasets, whose collection has significant costs and may be impractical. To address this issue, Transfer

Learning (TL) techniques are typically applied with the purpose of transferring and re-using knowledge across different data domains. In this setting, the simplest technique is to pre-train the model on a huge labeled dataset (*i.e.* the source) and then finetune it on the *target* task [78, 79, 80]. Such a simple schema has been recently overcome by more sophisticated domain adaptation algorithms [81, 82, 83] mainly based on the concept of *feature alignment*: here, the goal is to reduce the shift between the feature distributions of target and source domains. Unfortunately, these approaches often require the availability of the source dataset during training, which clashes with the usual constraints imposed in the CL scenarios.

In this work, we explore the interactions between pre-training and CL and highlight a blind spot of continual learners. Previous work underlined that naive pre-training is beneficial as it leads the learner to reduced forgetting [84]. However, we detect that the pre-training task itself is swiftly and catastrophically forgotten as the model veers towards the newly introduced stream of data. This matter is not really detrimental if all target classes are available at once (*i.e.*, joint training): 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 initialization. However, if classes are shown in a sequential manner, we argue that transfer mostly concerns the early encountered tasks: as a consequence, pre-training ends up being fully beneficial only for the former classes. For the later ones, since pre-training features are swiftly overwritten, the benefit of pre-training is instead lowered, thus undermining the advantages of the source knowledge. In support of this argument, this work reports several experimental analyses (Sec. 4.3.1) revealing that state-of-the-art CL methods do not take full advantage of pre-training knowledge.

## 4.2 Related Work

Continual Learning (CL) [18, 46] is an increasingly popular field of machine learning that deals with the mitigation of catastrophic forgetting [5]. CL methods are usually grouped as follows, according to the approach they take.

*Regularization-based* methods [23, 17, 35, 85] typically identify subsets of weights that are highly functional for the representations of previous tasks, with the purpose to prevent their drastic modification through apposite optimization constraints. Alternatively, they consolidate the previous knowledge by using past models as soft teachers while learning the current task [21].

*Architectural* approaches dedicate distinct sets of parameters to each task, often resorting to network expansion as new tasks arrive [26, 27, 28]. While capable of high performance, they are mostly limited to the Task-IL scenario (described in Sec. 4.4.1) as they require task-identifiers at inference time.

*Rehearsal-based* methods employ a fixed-size buffer to store a fraction of the old data. ER [6, 32] interleaves training samples from the current task with previous samples: notably, several works [68, 38] point out that such a simple strategy can effectively mitigate forgetting and achieve superior performance. This method has hence inspired several works: DER [38] and its extension X-DER [86] also store past model responses and pin them as an additional teaching signal. MER [70] combines replay and meta-learning [87, 88] to maximize transfer from the past while minimizing interference. Other works [34, 89] propose different sample-selection strategies to include in the buffer, while GEM [17] and its relaxation A-GEM [35] employ old training data to minimize interference. On

a final note, recent works [39, 90] exploit the memory buffer to address semi-supervised settings where examples can be either labeled or not.

**Transfer Learning (TL)** [9] is a machine learning methodology aiming at using the knowledge acquired on a prior task to solve a distinct target task. In its classical formulation [91], a model is trained on the source dataset and then finetuned on the (possibly much smaller) target dataset to adapt the previously learned features. Alternatively, transfer can be induced via multi-level Knowledge Distillation, guided by meta-learning [92], attention [93] or higher-level descriptions of the flow of information within the model [94].

## 4.3 Method

### Setting

In CL, a classification model $f_{(\theta,\phi)}$ (composed of a multilayered feature extractor $h_\theta = h^{(l)}_{\theta_l} \circ h^{(l-1)}_{\theta_{l-1}} \circ \cdots \circ h^{(1)}_{\theta_1}$ and a classifier $g_\phi$, $f_{(\theta,\phi)} = g_\phi \circ h_\theta$) is trained on a sequence of $N$ tasks $\mathcal{T}_i = \{(x^i_j, y^i_j)\}_{j=1}^{|\mathcal{T}_i|}$. The objective of $f_{(\theta,\phi)}$ is minimizing the classification error across all seen tasks:

$$\min_{\theta,\phi} \mathcal{L} = \mathop{\mathbb{E}}_{i} \left[ \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{T}_i} \Big[ \ell(y, f_{(\theta,\phi)}(x)) \Big] \right], \tag{4.1}$$

where $\ell$ is a suitable loss function. Unfortunately, the problem framed by Eq. 4.1 cannot be directly optimized due to the following key assumptions: *i)* while learning the current task $\mathcal{T}_c$, examples and labels of previous tasks are inaccessible; *ii)* the label space of distinct tasks is disjoint ($y^i_m \neq y^j_n \; \forall i \neq j$) *i.e.*, classes learned previously cannot recur in later phases. Therefore, Eq. 4.1 can only be approximated, seeking adequate

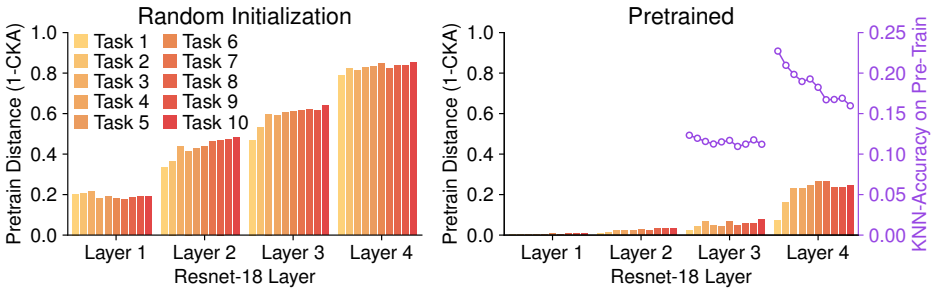



***Figure 4.1:*** *Forgetting of the initialization, measured as the distance from the pre-train ($1-CKA$ [95]) (lower is better) and $kNN$ accuracy (higher is better). Features extracted by a pre-trained model remain closer to the initialization w.r.t. a randomly initialized model. Furthermore, the steady decrease in $kNN$ accuracy as training progresses reveals that features become less specific for past tasks.*

performance on previously seen tasks (*stability*), while remaining flexible enough to adapt to upcoming data (*plasticity*).

### 4.3.1 Pre-training incurs Catastrophic Forgetting

*Mehta et al.* [84] have investigated the entanglement between continual learning and pre-training, highlighting that the latter leads the optimization towards wider minima of the loss landscape. As deeply discussed in [38, 86], such property is strictly linked to a reduced tendency in incurring forgetting.

On this latter point, we therefore provide an alternate experimental proof of the benefits deriving from pre-training initialization. In particular, we focus on ResNet-18 trained with ER [32] on Seq-CIFAR-100[1]

[1]This preliminary experiment follows the same setting presented in Sec. 4.4.1

and measure how each individual layer differs from its initialization. It can be observed that a randomly initialized backbone (Fig. 4.1, *left*) significantly alters its parameters at all layers while tasks progress, resulting in a very low Centered Kernel Alignment [95] similarity score already at the first CL task. On the contrary, a backbone pre-trained on Tiny ImageNet (Fig. 4.1, *right*) undergoes limited parameter variations in its layers, with the exception of the last residual layer (although to a lesser extent w.r.t. random init.). This latter finding indicates that its pre-training parametrization requires relevant modifications to fit the current training data. This leads 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 corroborated by the decreasing accuracy for pre-training data of a $k$NN classifier trained on top of *Layer 3* and *Layer 4* representations in Fig. 4.1 (*right*).

To sum up, while pre-training is certainly beneficial, the model drifts away from it one task after the other. Hence, only the first task takes full advantage of it; the optimization of later tasks, instead, starts from an initialization 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.

### 4.3.2 Transfer without Forgetting

To mitigate the issue above, we propose a strategy that enables a continuous transfer between the source task and the incrementally learned target problem.

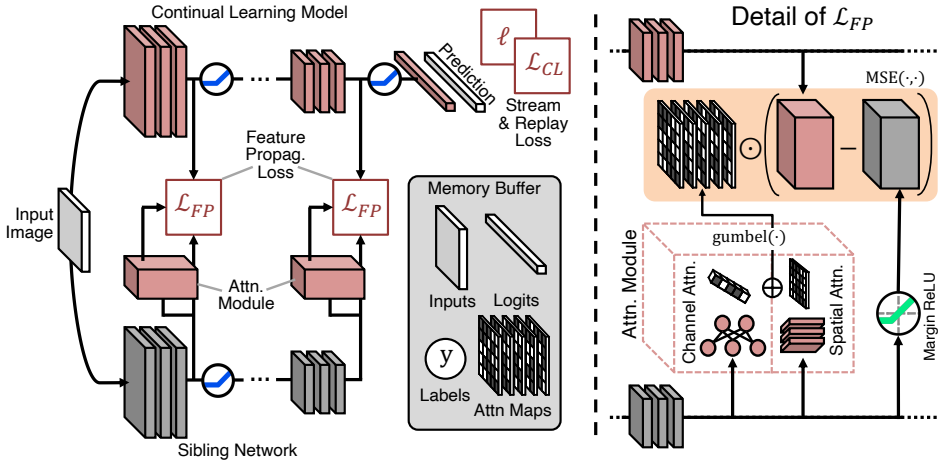



***Figure 4.2:*** *Overview of TwF and detail of $\mathcal{L}_{FP}$: Given a batch of samples from the current task or from $\mathcal{B}$, we i) extract intermediate features from both the student and fixed sibling backbones at multiple layers; ii) compute the corresponding binarized attention maps $\mathbb{M}(\cdot)$; iii) pull the attention-masked representations of the two models closer.*

### Feature Propagation

As the training progresses, the input stream introduces new classes that might benefit from the adaptation of specific features of the pre-trained model. To enable feature transfer without incurring pre-training forgetting, we maintain a copy of it (the *sibling* model) and adopt an intermediate feature knowledge distillation [96, 97, 93, 98, 99] objective. Considering a subset of $L$ layers, we seek to minimize the distance between the activations of the base network $h_\theta^{(l)} \triangleq h_\theta^{(l)}(x)$ and those from its pre-trained

sibling $\widehat{h}^{(l)} \triangleq h_{\theta^t}^{(l)}(x)$:

$$\mathop{\mathbb{E}}_{x \sim \mathcal{T}_c} \left[ \sum_{l=1}^{L} ||h_\theta^{(l)} - \mathrm{ReLU_m}(\widehat{h}^{(l)})||_2^2 \right], \tag{4.2}$$

where $c$ is the current task and $\mathrm{ReLU_m}(\cdot)$ indicates the application of a margin ReLU activation [98]. It is noted that the objective outlined by Eq. 4.2 leads the CL model to focus on mirroring the internal representations of the pre-trained teacher and maximizing transfer. However, focusing on the latter solely can lead to excessive rigidity, thus preventing the model from fitting the data from the current task altogether. On these grounds, we take inspiration from [93] and use a weighted version of Eq. 4.2. In particular, an apposite learnable module computes a gating attention map $\mathbb{M}(\cdot)$ over the feature maps of the sibling, which serves as a binary mask selecting which spatial regions have to be aligned. The resulting objective is consequently updated as follows:

$$\mathop{\mathbb{E}}_{x \sim \mathcal{T}_c} \left[ \sum_{l=1}^{L} ||\mathbb{M}(\widehat{h}^{(l)}) \odot \left( h_\theta^{(l)} - \mathrm{ReLU_m}(\widehat{h}^{(l)}) \right)||_2^2 \right], \tag{4.3}$$

where $\odot$ indicates the Hadamard product between two tensors of the same dimensions. The attention maps $\mathbb{M}(\cdot)$ are computed through specific layers, whose architectural design follows the insights provided in [100]. Specifically, they forward the input activation maps into two parallel branches, producing respectively a Channel Attention $\mathbb{M}_{\mathrm{Ch}}(\cdot)$ map and a Spatial Attention $\mathbb{M}_{\mathrm{Sp}}(\cdot)$ map. These two intermediate results are summed and then activated through a binary Gumbel-Softmax sampling [101], which allows us to model discrete *on-off* decisions regarding which information we want to propagate. In formal terms:

$$\mathbb{M}(\widehat{h}^{(l)}) \triangleq \mathrm{gumbel}(\mathbb{M}_{\mathrm{Ch}}(\widehat{h}^{(l)}) + \mathbb{M}_{\mathrm{Sp}}(\widehat{h}^{(l)})). \tag{4.4}$$

The Spatial Attention $\mathbb{M}_{\text{Sp}}(\widehat{h}^{(l)})$ regulates the propagation of spatially localized information and is obtained by stacking four convolutional layers [100] with different configurations:

$$\mathbb{M}_{\text{Sp}}(\widehat{h}^{(l)}) \triangleq \text{C}_{1\times1} \circ \text{C}_{3\times3} \circ \text{C}_{3\times3} \circ \text{C}_{1\times1}(\widehat{h}^{(l)}). \tag{4.5}$$

More in detail, $\mathbb{M}_{\text{Sp}}(\widehat{h}^{(l)})$ is computed on top of the activations of a given layer of the fixed sibling network $\widehat{h} \in \mathbb{R}^{b\times c\times h\times w}$, processed through a ResNet-inspired bottleneck structure [69, 100]:

$$\mathbb{M}_{\text{Sp}} \triangleq \text{C}^{\text{C}}_{1\times1} \circ \text{ReLU} \circ \text{BN} \circ \text{C}^{\text{B}}_{3\times3} \circ \text{ReLU} \circ \text{BN} \circ \text{C}^{\text{B}}_{3\times3} \circ \text{ReLU} \circ \text{BN} \circ \text{C}^{\text{A}}_{1\times1}, \tag{4.6}$$

where $\text{ReLU}$ denotes a ReLU activation, $\text{BN}$ indicates a Batch Normalization layer (conditioned on the task-identifier) and $\text{C}$ indicates a Convolutional layer. More specifically, $\text{C}^{\text{A}}_{1\times1}$ is a $1\times1$ convolution, projecting from $c$ channels to $c/4$; $\text{C}^{\text{B}}_{3\times3}$ is a $3\times3$ dilated convolution with dilation factor $2$ and adequate padding to maintain the same spatial resolution as the input, with $c/4$ channels both as input and output; $\text{C}^{\text{C}}_{1\times1}$ is a $1\times1$ convolution projecting from $c/4$ channels to $1$ channel. This results in $\mathbb{M}_{\text{Sp}}$ having shape $b\times1\times h\times w$.

On the other hand, the Channel Attention $\mathbb{M}_{\text{Ch}}(\widehat{h}^{(l)})$ estimates the information across the channels of $\widehat{h}^{(l)}$; in its design, we draw inspiration from the formulation proposed in [102]. Formally, considering the result $\widehat{h}^{(l)}_{\text{GAP}}$ of the Global Average Pooling (GAP) applied on top of $\widehat{h}^{(l)}$, we have:

$$\mathbb{M}_{\text{Ch}}(\widehat{h}^{(l)}) \triangleq \tanh(\text{BN}(W_1^{\mathsf{T}}\widehat{h}^{(l)}_{\text{GAP}})) \cdot \sigma(\text{BN}(W_2^{\mathsf{T}}\widehat{h}^{(l)}_{\text{GAP}})) + W_3^{\mathsf{T}}\widehat{h}^{(l)}_{\text{GAP}}, \tag{4.7}$$

where $W_1$, $W_2$, and $W_3$ are the weights of three fully connected layers organized in parallel and $\text{BN}$ indicates the application of batch normalization.

**Diversity loss**

Without a specific loss term supervising the attention maps, we could incur in useless behaviors, *e.g.*, all binary gates being either on or off, or some channels being always propagated and some others not. While recent works provide a target expected activation ratio [103, 28] as a countermeasure, we encourage the auxiliary modules to assign different propagation gating masks to different examples. The intuition is that each example has its own preferred subset of channels to be forwarded from the sibling. To do so, we include an additional auxiliary loss term [104] as follows:

$$\begin{aligned} \mathcal{L}_{\text{AUX}} &\triangleq -\lambda \sum_{l=1}^{L} \mathop{\mathbb{E}}_{x_1,\dots,x_n \sim \mathcal{T}_c} \left[ \sum_{j=1}^{n} \log \frac{e^{g_{ij}^{\mathrm{T}} g_{ij}/T}}{\frac{1}{n}\sum_{k=1}^{n} e^{g_{ij}^{\mathrm{T}} g_{ik}/T}} \right], \\ g_{ij} &\triangleq \text{NORM}(\text{GAP}(\mathbb{M}(\widehat{h}^{(l)}(x_j)))), \end{aligned} \tag{4.8}$$

where $n$ indicates the batch size, NORM a normalization layer, $T$ a temperature and finally $\lambda$ 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.

### 4.3.3 Knowledge Replay

The training objective of Eq. 4.3 is devised to facilitate selective feature transfer between the in-training model and the immutable sibling. However, to prevent forgetting tied to previous CL tasks to the greatest extent, the model should also be provided with a targeted strategy. We thus equip the continual learner with a small memory buffer $\mathcal{B}$ (populated with examples from the input stream via *reservoir sampling* [33]) and adopt the

simple labels and logits replay strategy proposed in [38]:

$$\mathcal{L}_{\text{CL}} \triangleq \mathop{\mathbb{E}}_{(x,y,l)\sim\mathcal{B}} \Big[ \alpha \cdot ||f_{(\theta,\phi)}(x) - l||_2^2 + \beta \cdot \ell(y, f_{(\theta,\phi)}(x)) \Big], \tag{4.9}$$

where $(x, y, l)$ is a triplet of example, label and original network responses $l = f(x)$ recorded at the time of sampling and $\alpha$, $\beta$ are scalar hyperparameters. Although extremely beneficial, we remark that the model need not optimize $\mathcal{L}_{\text{CL}}$ to achieve basic robustness against catastrophic forgetting (as shown in Sec. 4.5): preserving pre-training features already serves this purpose.

### Replaying past propagation masks

With the purpose of protecting the feature propagation formulated in Eq. 4.3 from forgetting, we also extend it to replay examples stored in memory. It must be noted that doing so requires taking additional steps to prevent cross-task interference; indeed, simply applying Eq. 4.3 to replay items would apply the feature propagation procedure unchanged to all tasks, regardless of the classes thereby included. For this reason, we take an extra step and make all batch normalization and fully connected layers in Eq. 4.4, 4.5 and 4.7 conditioned [105] w.r.t. the CL task. Consequently, we add to $\mathcal{B}$ 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. 4.3 is finally updated as:

$$\begin{aligned} \mathcal{L}_{\text{FP}} \triangleq & \mathop{\mathbb{E}}_{\substack{(x,t=c)\sim\mathcal{T}_c \\ (x;t)\sim\mathcal{B}}} \left[ \sum_{l=1}^{L} ||\mathbb{M}(\widehat{h}^{(l)}; t) \odot \Big( h^{(l)} - \text{ReLU}_{\text{m}}(\widehat{h}^{(l)}) \Big)||_2^2 \right] \\ & + \mathop{\mathbb{E}}_{\substack{(x,t,m)\sim\mathcal{B} \\ l=1,...,L}} \left[ \text{BCE}\Big( \mathbb{M}(\widehat{h}^{(l)}; t), m^{(l)} \Big) \right], \end{aligned} \tag{4.10}$$

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

### Overall objective

Our proposal – dubbed **Transfer without Forgetting (TwF)** – optimizes the following training objective, also summarized in Fig. 4.2:

$$\min_{\theta,\phi} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{T}_c} \left[\ell(y_j^i, f_{(\theta,\phi)}(x_j^i))\right] + \mathcal{L}_{\mathrm{CL}} + \mathcal{L}_{\mathrm{FP}} + \mathcal{L}_{\mathrm{AUX}}. \tag{4.11}$$

We remark that: *i)* while TwF requires keeping a copy of the pre-trained model during training, this does not hold at inference time; *ii)* similarly, task labels $t$ are not needed during inference but only while training, which makes TwF capable of operating under both the Task-IL and Class-IL CL settings [8]; *iii)* the addition of $t$ and $m$ in $\mathcal{B}$ 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 we demonstrate by storing $m$ at half resolution for our experiments in Sec. 4.4). We finally point out that, as maps $m$ take discrete binary values, one could profit from lossless compression algorithms (such as Run-Length Encoding [106] or LZ77 [107]) and thus store a compressed representation into the memory buffer. We leave the comprehensive investigation of this application to future works.

## 4.4 Experiments

### 4.4.1 Experimental Setting

#### Metrics

We assess the overall performance of the models in terms of *Final Average Accuracy* (FAA), defined as the average accuracy on all seen classes after learning the last task, and *Final Forgetting* [85] (FF), defined as:

$$\text{FF} \triangleq \frac{1}{T-1}\sum_{i=0}^{T-2} \max_{t\in\{0,\dots,T-2\}} \{a_i^t - a_i^{T-1}\}, \tag{4.12}$$

where $a_i^t$ denotes the accuracy on task $\tau_i$ after training on the $t^{\text{th}}$ task.

#### Settings

We report results on two common protocols [8]: *Task-Incremental Learning* (Task-IL), where the model must learn to classify samples only from within each task, and *Class-Incremental Learning* (Class-IL), where the model must gradually learn the overall classification problem. The former scenario is a relaxation of the latter, as it provides the model with the task identifier of each sample at test time; for this reason, we focus our evaluation mainly on the Class-IL protocol, highlighted as a more realistic and challenging benchmark [68, 34].

#### Datasets

We initially describe a scenario where the transfer of knowledge from the pre-train is facilitated by the similarity between the two distributions. Pre-

cisely, we use **CIFAR-100** [13] as the pre-train dataset and then evaluate the models on **Seq-CIFAR-10** [25] (5 binary tasks) (see Tab. 4.1). In Tab. 4.2 we envision a second and more challenging benchmark, which relies on **Seq-CIFAR-100** [25] with the opportunity to benefit from the knowledge previously learned on **Tiny-ImageNet** [15]. Due to the size mismatch between CIFAR-100 and the samples from Tiny ImageNet, we resize the latter to $32 \times 32$ during pre-training. The last scenario (Tab. 4.3) involves pre-training on ImageNet [108] and learning incrementally **Seq-CUB-200** [35, 109], split into 10 tasks of 20 classes each. With an average of only 29.97 images per class and the use of higher-resolution input samples (resized to $224 \times 224$), this benchmark is the most challenging. We use ResNet18 [69] for all experiments involving Seq-CIFAR-10 and Seq-CIFAR-100, as in [31, 38], while opting for ResNet50 on Seq-CUB-200.

### Competitors

We focus our comparison on state-of-the-art rehearsal algorithms, as they prevail on most benchmarks in literature [47, 38, 8].

- **Experience Replay (ER)** [6, 32] is the first embodiment of a rehearsal strategy that features a small memory buffer containing an *i.i.d.* view of all the tasks seen so far. During training, data from the stream is complemented with data sampled from the buffer. While this represents the most straightforward use of a memory in a CL scenario, ER remains a strong baseline, albeit with a non-negligible memory footprint.

- **Dark Experience Replay (DER)** [38] envisions a self-distillation [110] constraint on data stored in the memory buffer and

represents a simple extension to the basic rehearsal strategy of ER. In this work, we compare against DER++, which includes both ER and DER objectives.

- **Incremental Classifier and Representation Learning (iCaRL)** [31] tackle catastrophic forgetting by distilling the responses of the model at the previous task boundary and storing samples that better represent the current task. In addition to simple replay, those *exemplars* are used to compute class-mean prototypes for nearest-neighbor classification.

- **ER with Asymmetric Cross-Entropy (ER-ACE)** [40] recently introduced a method to alleviate class imbalances to ER. The authors obtain a major gain in accuracy by simply separating the cross-entropy contribution of the classes in the current batch and that of the ones in the memory buffer.

- **Contrastive Continual Learning (CO$^2$L)** [41] proposes to facilitate knowledge transfer from samples stored in the buffer by optimizing a contrastive learning objective, avoiding any potential bias introduced by a cross-entropy objective. To perform classification, a linear classifier needs to be first trained on the exemplars stored in the buffer.

In addition, we also include results from two popular regularization methods. **Online Elastic Weight Consolidation (oEWC)** [23] penalizes changes on the most important parameters by means of an online estimate of the Fisher Information Matrix evaluated at task boundaries. **Learning without Forgetting (LwF)** [21] includes a distillation target similar to iCaRL but does not store any exemplars. We remark that **all competitors**

| **FAA (FF)** | **Seq-CIFAR-10** (*pretr. CIFAR-100*) | | | |
|---|---|---|---|---|
| Method | *Class-IL* | | *Task-IL* | |
| Joint | 92.89 (–) | | 98.38 (–) | |
| Finetune | 19.76 (98.11) | | 84.05 (17.75) | |
| oEwC [24] | 26.10 (88.85) | | 81.84 (19.50) | |
| LwF [21] | 19.80 (97.96) | | 86.41 (14.35) | |
| Buffer Size | 500 | 5120 | 500 | 5120 |
| ER [32] | 67.24 (38.24) | 86.27 (13.68) | 96.27 (2.23) | 97.89 (0.55) |
| $CO^2L$ [41] | 75.47 (21.80) | 87.59 (9.61) | 96.77 (1.23) | 97.82 (0.53) |
| iCaRL [31] | 76.73 (14.70) | 77.95 (12.90) | 97.25 (0.74) | 97.52 (0.15) |
| DER++ [38] | 78.42 (20.18) | 87.88 (8.02) | 94.25 (4.46) | 96.42 (1.99) |
| ER-ACE [40] | 77.83 (10.63) | 86.20 (5.58) | 96.41 (2.11) | 97.60 (0.66) |
| **TwF (ours)** | **83.65**(**11.59**) | **89.55**(**6.85**) | **97.49**(**0.86**) | **98.35**(**0.17**) |

***Table 4.1:*** *Final Average Accuracy (FAA) [↑] and Final Forgetting (FF) [↓] on Seq-CIFAR-10 w. pre-training on CIFAR-100.*

**undergo an initial pre-training phase** prior to CL, thus ensuring a fair comparison.

To gain a clearer understanding of the results, all the experiments include the performance of the upper bound (**Joint**), obtained by jointly training on all classes in a non-continual fashion. We also report the results of the model obtained by training sequentially on each task (**Finetune**), *i.e.*, without any countermeasure to forgetting.

| **FAA (FF)** | **Seq-CIFAR-100** (*pretr. Tiny-ImageNet*) | | | |
|---|---|---|---|---|
| Method | *Class-IL* | | *Task-IL* | |
| Joint (UB) | 75.20 (–) | | 93.40 (–) | |
| Finetune | 09.52 (92.31) | | 73.50 (20.53) | |
| oEwC [24] | 10.95 (81.71) | | 65.56 (21.33) | |
| LwF [21] | 10.83 (90.87) | | 86.19 (4.77) | |
| Buffer Size | 500 | 2000 | 500 | 2000 |
| ER [32] | 31.30 (65.40) | 46.80 (46.95) | 85.98 (6.14) | 87.59 (4.85) |
| CO$^2$L [41] | 33.40 (45.21) | 50.95 (31.20) | 68.51 (21.51) | 82.96 (8.53) |
| iCaRL [31] | 56.00 (19.27) | 58.10 (16.89) | **89.99(2.32)** | 90.75 (1.68) |
| DER++ [38] | 43.65 (48.72) | 58.05 (29.65) | 73.86 (20.08) | 86.63 (6.86) |
| ER-ACE [40] | 53.38 (21.63) | 57.73 (17.12) | 87.21 (3.33) | 88.46 (2.46) |
| **TwF (ours)** | **56.83(23.89)** | **64.46(15.23)** | 89.82 (3.06) | **91.11(2.24)** |

***Table 4.2:*** *Final Average Accuracy (FAA) [↑] and Final Forgetting (FF) [↓] on Seq-CIFAR-100 w. pre-training on Tiny-ImageNet.*

### 4.4.2 Comparison with State-Of-The-Art

#### Regularization methods

Across the board, non-rehearsal methods (oEWC and LwF) manifest a profound inability to effectively use the features learned during the pre-train. As those methods are not designed to extract and reuse any useful features from the initialization, the latter is rapidly forgotten, thus negating any knowledge transfer in later tasks. This is particularly true for oEWC,

| **FAA (FF)** | **Seq-CUB-200** (*pretr. ImageNet*) | | | |
|---|---|---|---|---|
| Method | *Class-IL* | | *Task-IL* | |
| Joint (UB) | 78.54 (–) | | 86.48 (–) | |
| Finetune | 8.56 (82.38) | | 36.84 (50.95) | |
| oEwC [24] | 8.20 (71.46) | | 33.94 (40.36) | |
| LwF [21] | 8.59 (82.14) | | 22.17 (67.08) | |
| Buffer Size | 400 | 1000 | 400 | 1000 |
| ER [32] | 45.82 (40.76) | 59.88 (25.65) | 75.26 (9.82) | 80.19 (4.52) |
| $CO^2L$ [41] | 8.96 (32.04) | 16.53 (20.99) | 22.91 (26.42) | 35.79 (16.61) |
| iCaRL [31] | 46.55 (12.48) | 49.07 (11.24) | 68.90 (3.14) | 70.57 (3.03) |
| DER++ [38] | 56.38 (26.59) | 67.35 (13.47) | 77.16 (7.74) | 82.00 (3.25) |
| ER-ACE [40] | 48.18 (25.79) | 58.19 (16.56) | 74.34 (9.78) | 78.27 (6.09) |
| **TwF (ours)** | **57.78**(18.32) | **68.32**(6.74) | **79.35**(5.77) | **82.81**(2.14) |

***Table 4.3:*** *Final Average Accuracy (FAA) [↑] and Final Forgetting (FF) [↓] on CUB-200 w. pre-training on ImageNet.*

whose objective proves to be both too strict to effectively learn the current task and insufficient to retain the initialization. Most notably, on Seq-CUB-200 oEWC shows performance lower than Finetune on both Task- and Class-IL.

### Rehearsal methods

In contrast, rehearsal models that feature some form of distillation (DER++ and iCaRL) manage to be competitive on all benchmarks. In particular,

iCaRL proves especially effective on Seq-CIFAR-100, where it reaches 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 Seq-CIFAR-10 and Seq-CUB-200, where the role of pre-training is far more essential due to the similarity of the two distributions for the former and the higher 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 initialization. Moreover, we remark that iCaRL and DER++ show ranging Class-IL performance in different tasks, whereas our method is much less sensitive to the specific task at hand.

While it proves effective on the easier Seq-CIFAR-10 benchmark, CO$^2$L does not reach satisfactory results on either Seq-CIFAR-100 or Seq-CUB-200. 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). Remarkably, while we extended the size of the batch in all experiments with CO$^2$L to $256$ to provide a large enough pool of negative samples, it still shows off only a minor improvement on non-rehearsal methods for Seq-CUB-200.

Interestingly, while both ER and ER-ACE do not feature distillation, we find their performance to be competitive for large enough buffers. In particular, the asymmetric objective of ER-ACE appears less sensitive to a small memory buffer but always falls short of DER++ when this constraint is less severe.

**Transfer without Forgetting**

Finally, results across all proposed benchmarks depict our method (TwF) as consistently outperforming all the 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 Seq-CIFAR-10 and Seq-CUB-200, for which the pre-train provides a valuable source of knowledge to be transferred. We argue that this proves the efficacy of our proposal to retain and adapt features available from initialization through distillation. Moreover, we remark that its performance gain is consistent in all settings, further attesting to the resilience of the proposed approach.

## 4.5 Ablation Studies

**Breakdown of the individual terms of TwF**

To better understand the importance of the distinct loss terms in Eq. 4.11 and their connection, we explore their individual contribution to the final accuracy of TwF in Tab. 4.4. Based on these results, we make the following observations: *i)* $\mathcal{L}_{\text{CL}}$ 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 $\mathcal{L}_{\text{CL}}$ 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 labeled as $^{\text{w/o}}/_{\text{buf.}}$, we consider what happens if TwF is allowed **no replay example at all** and only optimizes $\mathcal{L}_{\text{FP}}$ and

| $\mathcal{L}_{\text{CL}}$ | $\mathcal{L}_{\text{FP}}$ | $\mathcal{L}_{\text{AUX}}$ | **Seq-CIFAR-10** | | | **Seq-CIFAR-100** | | | **Seq-CUB-200** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Buffer Size | | | 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 |

***Table 4.4:*** *Impact of each loss term and of using no memory buffer on TwF. Results given in the Class-IL scenario following the same experimental settings as Tab.4.1-4.3.*

$\mathcal{L}_{\text{AUX}}$ on current task examples. Compared to oEWC in Tab. 4.1-4.3 – 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 is out of the picture.

**Alternatives for the preservation of pre-training knowledge**

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.

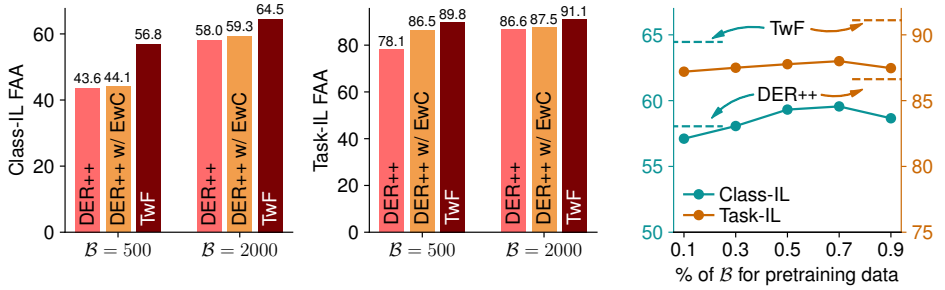



***Figure 4.3:*** *Class-IL (left) and Task-IL (center) FAA performance comparison of our proposal with different possible methods to retain knowledge from pre-train. (Right) Influence of different allocation rates of pre-train examples in $\mathcal{B}$ for DER++, $|\mathcal{B}| = 2000$.*

### Pre-training preservation with EWC

We complement a strong approach such as DER++ with an additional regularization term based on EWC:

$$\mathcal{L}_{\text{EWC}} = \lambda(\theta - \theta^t)^T \operatorname{diag}(F)(\theta - \theta^t), \tag{4.13}$$

where $\operatorname{diag}(F)$ indicates the diagonal of the empirical Fisher Information Matrix, estimated on the pre-training data at the optimum $\theta^t$. When equipped with this additional loss term, DER++ is anchored to its initialization 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. 4.3 (left, center), the EWC loss allows DER++ to improve its accuracy on Seq-CIFAR-100 with Tiny ImageNet pre-training (especially in the Task-IL setting). However, this improvement is not actively incentivizing 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 initialization checkpoint).

### Pre-training preservation through rehearsal

An alternative for preserving the source knowledge is to assume that pre-training data is available and can be treated as an auxiliary data stream [111]. To evaluate this strategy with a bounded memory footprint, we test our baseline method (DER++) on Seq-CIFAR-100 with different percentages of the buffer dedicated to pre-training images (from Tiny ImageNet). The results shown in Fig. 4.3 (right) confirm our main 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.

Furthermore, we highlight that TwF outperforms the baseline introduced here. When replaying pre-training data, indeed, the model has 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 favors the learning of new tasks ($\Rightarrow$ **it only enforces forward transfer**). This substantial distinction helps to understand the merits of TwF: namely, a full but still functional exploitation of the pre-training knowledge.

### Role of pre-training datasets

Here, we seek to gain further proof of our claim about the ability of TwF to adapt features from the pre-train. Specifically, we study a scenario where the source data distribution and the target one are highly dissimilar: namely, we first pre-train a ResNet18 backbone on SVHN [112] and then

| **FAA (FF)** | *Class-IL* | | *Task-IL* | |
|---|---|---|---|---|
| Buffer size | 500 | 2000 | 500 | 2000 |
| iCaRL [31] | 39.59 (21.81) | 42.02 (18.78) | 78.89 (4.04) | 80.65 (2.24) |
| DER++ [38] | 36.46 (53.47) | 52.29 (24.04) | 75.05 (16.22) | 83.36 (8.04) |
| **TwF (ours)** | **43.56**(40.02) | **56.15**(21.51) | **80.89**(10.12) | **87.30**(3.12) |

***Table 4.5:*** *Dissimilar pre-training tasks: (FAA) [↑] and Final Forgetting (FF) [↓] on Seq-CIFAR-100 pre-trained on SVHN.*

follow with Seq-CIFAR-100. We compare our model with the second-best performer from Tab. 4.2, *i.e.*, iCaRL, and DER++. The results, reported in Tab. 4.5, suggest that our method outranks the competitors not only when pre-trained on a similar dataset – as in Tab. 4.2 – 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 are really advantageous to transfer.

## 4.6 Discussion

We introduced Transfer without Forgetting, a hybrid method combining Rehearsal and Feature transfer, designed to exploit pre-trained weights in an incremental scenario. It encourages feature sharing throughout all tasks, yielding a stable performance gain across multiple settings. We also show that TwF outperforms other hybrid methods based on rehearsal and regularization and that it is able to profit even from pre-training on a largely dissimilar dataset.

## 4.7 Publications

Boschini, M., Bonicelli, L., Porrello, A., Bellitto, G., Pennisi, M., Palazzo, S., Spampinato, C., Calderara, S. (2022). Transfer without forgetting. In European Conference on Computer Vision (pp. 692-709). Cham: Springer Nature Switzerland.

CHAPTER

# FIVE

# EFFECTIVENESS OF EQUIVARIANT REGULARIZATION IN CONTINUAL LEARNING

Chapter 3 and Chapter 4 have delved into two alternative ways to exploit past knowledge for continuously solving new tasks. However, the proposed methods have one aspect in common: they both rely on the use of external data (used in conjunction with the primary data stream in one case, or as source data employed during the pre-training of the sibling network in the other). In this chapter, we attempt to overcome this ”limitation” by removing the need of additional data beyond the stream one.

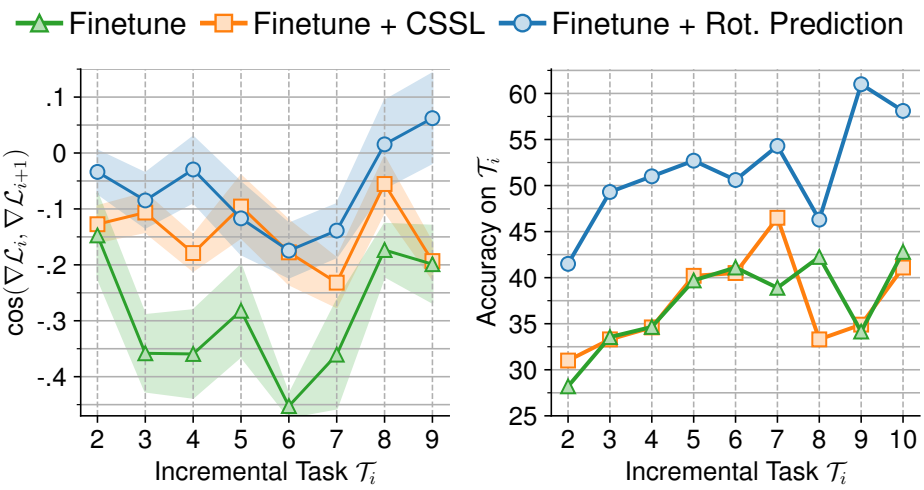
Another potential way we investigate to leverage auxiliary knowledge is by coupling the continual classification task with a different auxiliary task that guides learning. However, the main assumption to make this approach effective is a *forgetting-free* behaviour. This means that the surrogate task should act on a ideally *i.i.d.* data within the context of continual

learning. In Self-Supervised Learning (SSL), it has been demonstrated that a pretext task allows for learning a semantically good representation that may impact positively performance of downstream task [113]. In this chapter we propose to leverage equivariant tasks, as a form of self-supervision, and verify that they are particularly effective in extreme scenario such as the Online Continual Learning (OCL), where only one iteration of the input dataset is permitted.

## 5.1 Motivation

Motivated by the success of Contrastive Self-Supervised Learning (CSSL) [114, 115, 116], several recent CL approaches pivot on self-supervised representation learning [43, 41, 117, 118]. Indeed, as self-supervised representations are generally acknowledged to be agnostic and easily transferable to diverse downstream tasks [81], their exploitation appears especially promising in the online scenario, where learning a shared representation across tasks is as important as the prevention of forgetting. Moreover, we argue that binding the incoming classes to general-purpose representations encourages the emergence of a horizontal and shareable knowledge base, that will be less subject to forgetting.

However, we reckon that the CSSL paradigm is not a silver bullet: indeed, contrastive learning methods are characterized by low *sample efficiency* as their convergence requires large amounts of resources. As a result, CL methods need a higher number of training epochs when equipped with contrastive regularization [41], which clashes with the constraints of OCL. Moreover, they usually focus their representation learning on a small memory buffer [43], which entails a high risk of overfitting [119].




(a) (b)

***Figure 5.1:*** *Effects of SSL in OCL (Seq-CIFAR-100) comparing a Finetuning baseline with no additional regularization (green), with a Contrastive SSL auxiliary objective (orange) and with an Equivariant rotation prediction pretext task (blue). (a) Similarity between the gradients induced on the model by task $\mathcal{T}_i$ and $\mathcal{T}_{i+1}$ after training on $\mathcal{T}_i$. (b) Accuracy on task $\mathcal{T}_i$ after training on $\mathcal{T}_i$. Results are reported after a warm-up task (best in colors).*

This work addresses these limitations, revealing the benefits of *equivariant* self-supervised tasks (*e.g.*, rotation prediction, jigsaw puzzle, ...) for the OCL scenario. To provide an insight, Fig. 5.1 considers a simple learner based on Finetuning (*i.e.*, no counter-measure against forgetting) and reports its performance in the online scenario allowing only one epoch per task: in doing so, we compare the effects of the auxiliary objective based either on equivariant self-supervised learning (in this case, four-fold

rotation prediction) or on Barlow Twins [115], a recent CSSL-based approach that has also shown its merit in CL [43]. We observe that both representation learning tasks allow for a lower interference between features learned by SSL, as supported by the more favorable alignment of gradients between current and subsequent tasks (Fig. 5.1a). Surprisingly, Fig. 5.1b shows that only the rotation-aided model has a significant profit in terms of individual task accuracy for the CSSL-based objective. We conjecture that the limited amount of training steps in online CL is not sufficient for contrastive approaches (such as Barlow Twins) to produce effective representations for the downstream task.

To address the aforementioned CSSL limitations in the OCL setting, we propose **Continual Learning via Equivariant Regularization (CLER)**, a novel OCL regularizer built on top of equivariant pretext tasks – to the best of our knowledge, this is the first attempt to exploit equivariant information in CL. We demonstrate that our proposal can be easily combined with existing state-of-the-art CL approaches, leading to a generalized improvement in performance. Through additional experiments, we highlight the structural and predictive properties conferred by CLER and draw a detailed comparison with CSSL-based alternatives.

## 5.2 Related Work

**(Online) Continual Learning** is a field of machine learning that studies training over sequences of non-i.i.d. tasks, with the objective of retaining as much knowledge as possible from older tasks and mitigating catastrophic forgetting [5]. The existing literature offers different techniques to tackle this problem: *regularization-based* [23, 21] methods are designed

to control parameter updates in order to prevent disruptive modifications to features important for previous tasks; *segregation-based* [27, 28] approaches identify subsets of task-relevant parameters and prevent their alteration by combining parameter freezing, model expansion, and feature gating; *replay-based* [32, 70, 38, 40] methods store examples from the past in a memory buffer, with the objective of periodically refreshing older knowledge. Despite its simplicity, the latter approach is usually regarded as the most effective solution to date [68, 8, 47].

These methods are typically evaluated in a relaxed training setting, where the current task can be experienced over multiple epochs. In practical applications, this requirement is rarely satisfied; Online CL (OCL) [120, 17, 34] is a challenging and realistic scenario that adds the condition that each sample of the stream can be seen only once. Works targeting OCL typically all belong to the *replay-based* family [17, 47][1]. Among recent proposals, MIR [62] and GSS [34] propose enhanced replay sample selection procedures, ER-AML/ER-ACE [40] encourage balance in learning by means of carefully designed loss functions, CoPE [121] learns by exploiting slowly evolving class summaries.

**Self-Supervised Representation Learning in CL.** Self-Supervised Learning aims at learning useful representations directly from the data, *i.e.*, with no need for manual annotations. Recent SSL works show that these methods are able to learn strong representations that can reach or even outperform those of supervised learning [81, 114, 115]. In the context of CL, SSL methods are typically trained to encourage the backbone network to be invariant to the given transformations [41, 117, 43, 118, 66]. Co$^2$L [41] learns the representations for new tasks with a modified super-

[1] All contemporary OCL works consider only replay approaches, due to their clear performance superiority over all alternatives [120, 40].

vised contrastive learning procedure [42], where current task samples are used as anchors and elements in the buffer are used as negative samples – all this while preserving past knowledge through distillation. However, applying SSL methods in CL is not straightforward: SSL benefits from large batch sizes and require several training steps to converge [81]; this represents a limit for Co$^2$L, as the number of negative samples is limited by the small buffer size. DualNet [43] decouples representation learning from the CL objective through two complementary networks: a *slow net* exploits buffer samples to learn an overall representation, while a *fast net* sequentially learns from the input stream, using the features from the slow net to guide the process.

**Pretext Self-Supervised Learning and Rotations.** Differently from CSSL, [122] employs a *four-fold rotation* prediction pretext task to provide a powerful learning signal for representation learning. In [123], the rotation pretext task is applied in the context of few-shot learning; similarly, [124] pairs rotation prediction to existing SSL methods, leading to a consistent performance improvement. Recently, the authors of [125] investigated the role of invariance and equivariance in SSL, suggesting that some transformations (*e.g.*, four-fold rotations, jigsaw puzzle) can be effective when employed to encourage equivariance, but can lead to disruptive effects when enforcing invariance.

## 5.3 Method

### 5.3.1 Online Continual Learning

In Online Continual Learning (OCL) [34, 126], a single DNN $f_\theta$ is trained on a sequence of classification tasks $\mathcal{T}_1, \dots, \mathcal{T}_T$. Each task consists of dis-

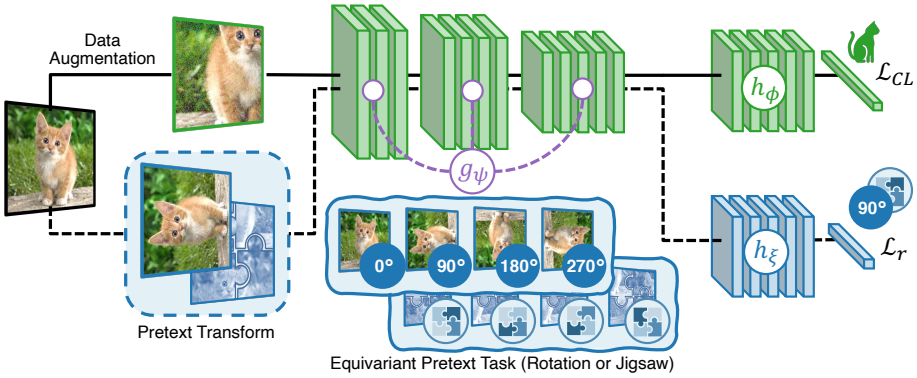



***Figure 5.2:*** *Overview of CLER. Two versions of the input image are fed into the in-training model: i) standard data augmentation is used to train the classification head (green); ii) an equivariant transformation-based task (rotation, alternatively jigsaw) is used to train the pretext head (blue).*

joint input and output distributions ($\mathcal{T}_i = (\mathcal{X}_i, \mathcal{Y}_i)$, with $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$) and each example-label pair may only be shown to the model once. At task $\mathcal{T}_c$, CL aims at optimizing $f_\theta$ on all $T$ tasks, while only having access to data from $\mathcal{T}_c$ itself:

$$\mathcal{L} = \sum_{i=1}^{T} \mathcal{R}_i = \underbrace{\sum_{i=1}^{c-1} \mathcal{R}_i}_{\substack{① \\ \text{data no longer} \\ \text{available}}} + \underbrace{\mathcal{R}_c}_{\substack{② \\ \text{data available}}} + \underbrace{\sum_{j=c+1}^{T} \mathcal{R}_j}_{\substack{③ \\ \text{data not yet} \\ \text{available}}}, \tag{5.1}$$

where $\mathcal{R}_i = \mathbb{E}_{(x,y)\in\mathcal{T}_i}\big[\ell(f_\theta(x), y)\big]$ denotes the empirical risk associated with the data of task $\mathcal{T}_i$.

In Eq. 5.1, term ① (stability) requires $f_\theta$ to maintain predictive efficacy on previously encountered data, whereas term ③ (plasticity) suggests that the model should prepare for fitting novel data distributions in later tasks.

Only ② can be directly pursued by training on data; instead, ① and ③ are achieved by means of auxiliary loss terms. CL methods endeavor to balance the three terms, which are typically understood to interfere with one another [70, 127, 128].

### 5.3.2 OCL via Equivariant Regularization

The objectives ① and ③ from Eq. 5.1 characterize the main challenges that come when designing a CL model. However, both can be addressed by learning a representation that can be shared across multiple tasks. To achieve this, we equip the online learner with an auxiliary SSL objective. Works in current literature pursue this objective through CSSL loss terms [41, 43]; instead, we follow the insights presented in Sec. 5.1 and opt for an *equivariant* pretext task [124], defined as follows.

Let $\mathcal{A} = \{\mathcal{A}_i\}_{i=1}^{K}$ be a family of input transforms $\mathcal{A}_i : \mathcal{X} \to \mathcal{X}$ (*e.g.*, rotations, jigsaw puzzle), we transform each input exemplar with a randomly chosen $\mathcal{A}_k$ and request the in-training model to recognize the transformation by predicting the correct label $k \in \mathcal{Y}_{\mathcal{A}} = \{1, \dots, K\}$. For this purpose, we rewrite $f_\theta$ as $h_\phi \circ g_\psi$, where $g_\psi$ is the early part of the network, devoted to the extraction of features, and $h_\phi$ encompasses the latter part of the model, including the final multi-layer classification head for the CL task. Subsequently, we introduce $h_\xi$: a separate sub-network following the same structure as $h_\phi$, finally projecting the representation $g_\psi(\cdot)$ on the set $\mathcal{Y}_{\mathcal{A}}$.

We treat the choice of $\mathcal{A}$ as a hyperparameter. In our experiments, we explore two different kinds of transformations: the set of $4$ non-distorting image rotations $\{\mathrm{Rot}_{0^\circ}, \mathrm{Rot}_{90^\circ}, \mathrm{Rot}_{180^\circ}, \mathrm{Rot}_{270^\circ}\}$ [123, 122], and the $24$ permutations of patches produced by a $2 \times 2$ jigsaw puzzle [129]. The

resulting approach, called CLER, consists of a regularization term $\mathcal{L}_r$ that can be readily applied on a backbone network as shown in Fig. 5.2. Let $\mathbf{x} \in \mathbf{B}_{\text{in}}$ be a sample coming from the input batch, we define $\mathcal{L}_r$ as:

$$\mathcal{L}_r = \lambda_r \cdot \mathop{\mathbb{E}}_{\substack{\mathbf{x}\sim\mathbf{B}_{\text{in}} \\ k\sim\mathcal{Y}_{\mathcal{A}}}} \left[ \text{CE}\left( h_\xi(g_\psi(\mathcal{A}_k(\mathbf{x}))), k \right) \right], \tag{5.2}$$

where CE is the cross-entropy loss and $\lambda_r$ is a scalar hyper-parameter to control the strength of the regularization. We highlight that the label space $\mathcal{Y}_{\mathcal{A}}$ of the pretext task remains constant over time. The objective of CLER can hence be compared to classification problems where only the data-generating distribution is subject to changes (Domain-Incremental learning [8]).

### Equivariance & invariance

A function $f_\theta$ is said to be equivariant w.r.t. $\mathcal{A}$ if there exists a mapping $\mathcal{M}_{\mathcal{A}}$ such that:

$$f_\theta(A(\mathbf{x})) = \mathcal{M}_{\mathcal{A}}(f_\theta(\mathbf{x})), \quad \forall \mathbf{x} \in \mathcal{X}. \tag{5.3}$$

While the learning objective in Eq. 5.2 promotes sensitivity to the chosen set of transformations, solving the CL task forces the model to become invariant w.r.t. employed data augmentations. To avoid overlapping between the two objectives, we compute Eq. 5.2 only on non-augmented inputs.

## 5.4 Experiments

### 5.4.1 Experimental setting

#### Benchmarks

We build our OCL benchmarks by taking image classification datasets and splitting their classes equally into a series of disjoint tasks. In the online learning scenario, the learner will then experience each task **only once** (single epoch). For additional details regarding the experiments, we refer the reader to the supplementary material.

- **Seq-CIFAR-100** [25, 31, 47] is obtained by splitting the original $100$ classes of CIFAR-100 [13] into $10$ consecutive tasks. For each class, train and test sets include $500$ and $100$ $32 \times 32$ RGB images respectively.
- **Seq-Mini-ImageNet** [47, 130, 131] is a challenging dataset that includes a total of $100$ classes from the popular ImageNet dataset and a longer sequence of tasks. While the number of samples is the same as in Seq-CIFAR-100, images are resized to $84 \times 84$ and split into $20$ $5$-way tasks.

#### Evaluation protocol

We primarily focus our evaluation on the online Class-Incremental (oCIL) setting, where the model is asked to gradually learn to solve all tasks, with no information regarding the task identifier (Task-ID). Differently from the online Task-Incremental (oTIL) setting, where the task Task-ID is available during inference, oCIL forces the learner to build a single-headed classifier. We present extensive results in both the oCIL and oTIL settings.

### Baseline methods

We report the results of CLER on a selection of current state-of-the-art (SOTA) methods viable for the oCIL setting.

- **Experience Replay with Asymmetric Cross-Entropy (ER-ACE)** [40]. Starting from the popular store-and-replay baseline (Experience Replay [6, 32]), the authors propose an alteration aimed at preventing imbalances due to the simultaneous optimization of current and past data.
- **eXtended Dark Experience Replay (X-DER)** [86] is a model that combines replay with self-distillation, while adopting careful design choices to harmonically blend predictive functions learned at different times.
- **Continual Prototype Evolution: Learning Online from Non-Stationary Data Streams (CoPE)** [121] proposes a classifier based on class prototypes, whose careful update scheme allows for learning incrementally while avoiding sudden disruptions in the latent space.
- **DualNet** [43] is a dual-backbone architecture decoupling the issue of incremental classification from the one of learning an overall transferable representation. The latter task is demanded to one of the backbones (*slow learner*), trained with a CSSL loss term on i.i.d. data coming from the replay buffer; the other backbone (*fast learner*) is instead tasked with fitting the CL tasks while taking advantage of the representations produced by the slow learner.

All models are trained for a single epoch with SGD, with a fixed batch size of $10$ both on the input stream and the replay buffer. We benchmark all models with two different sizes for the memory buffer: $500$ and $2000$ for Seq-CIFAR-100 and $2000$ and $8000$ for Seq-Mini-ImageNet. For these methods the input $\mathrm{B}_{\text{in}}$ in Eq. 5.2 is the concatenation of the images coming both from the stream and the buffer.

| **oCIL** | **Seq-CIFAR-100** | | **Seq-Mini-ImageNet** | |
|---|---|---|---|---|
| Joint-offline | 69.47 (–) | | 63.31 (–) | |
| Joint-online | 23.14 (–) | | 10.68 (–) | |
| Finetune | 7.00 (100) | | 3.21 (100) | |
| **Buffer Size** | 500 | 2000 | 2000 | 8000 |
| ER-ACE [40] | 20.17 (38.75) | 26.95 (23.69) | 15.03 (35.01) | 16.07 (37.94) |
| + CLER | **24.53**$^{JS}$ (33.76) | **30.89**$^{JS}$ (20.24) | **18.08**$^{R}$ (32.53) | **18.43**$^{JS}$ (33.22) |
| X-DER [86] | 25.80 (39.54) | 30.44 (31.52) | 17.51 (34.25) | 18.01 (50.84) |
| + CLER | **29.35**$^{JS}$ (35.56) | **34.57**$^{JS}$ (29.71) | **21.26**$^{JS}$ (34.07) | **21.71**$^{JS}$ (34.76) |
| CoPE [121] | 19.98 (75.32) | 34.09 (46.39) | 22.67 (57.96) | 24.54 (55.09) |
| + CLER | **26.15**$^{JS}$ (69.28) | **38.48**$^{JS}$ (45.50) | **25.91**$^{R}$ (57.73) | **26.76**$^{R}$ (52.69) |
| DualNet [43] | 11.09 (92.42) | 19.93 (73.44) | 16.21 (80.35) | 25.33 (59.60) |
| + CLER | **11.89**$^{R}$ (89.97) | **20.88**$^{JS}$ (73.02) | **18.66**$^{R}$ (72.74) | **30.90**$^{R}$ (52.14) |

***Table 5.1:*** *Final Average Accuracy (FAA) (↑) and Final Average Adjusted Forgetting ($\bar{F}^*_F$) (↓) on the oCIL setting.* $^{R}$ *indicates a result obtained with rotation,* $^{JS}$ *a result obatined with* $2 \times 2$ *jigsaw puzzle.*

To better compare the effect of CLER, we also include the results of a model jointly trained on all classes for one epoch (**Joint-online**) and for 30 and 50 epochs respectively on Seq-CIFAR-100 and Seq-Mini-ImageNet (**Joint-offline**). Also, we include the results of a model trained on the task sequence with no forgetting countermeasures (**Finetune**).

Architecture We rely on ResNet18 [69] as backbone in all experiments. For

DualNet, we use this model as the slow learner and – in line with [43] – construct the fast learner as a feed-forward network with the same number of convolutional layers as residual blocks in the slow learner.

Regardless of the underlying CL method, we define the feature extractor $g_\phi$ and the classification heads $h_\phi$ and $h_\xi$ by splitting the ResNet backbone at the second-last residual block; namely, $h_\phi$ and $h_\xi$ are comprised of the last residual block, followed by a linear projection onto the respective sets of classes $\mathcal{Y} = \cup_{i=1}^{T} \mathcal{Y}_i$ and $\mathcal{Y}_{\mathcal{A}}$.

### Metrics

As a primary indicator of a model's performance at the end of OCL, we report its *Final Average Accuracy* (FAA). Let $a_i^j$ be the accuracy of the model at the end of task $j$ computed on the test set of task $\mathcal{T}_i$, FAA is computed as:

$$FAA = \frac{1}{T} \sum_{i=1}^{T} a_i^T. \tag{5.4}$$

To further assess learning as tasks progress, we report the *Final Average Adjusted Forgetting* ($\bar{F}^*_F$), defined as follows:

$$\begin{aligned} \bar{F}^*_F &= \frac{1}{T-1} \sum_{i=1}^{T-1} \left[ \frac{a_i^* - a_i^T}{a_i^*} \right]^+, \\ \text{where } \; a_i^* &= \max_{t \in \{i,\dots,T-1\}} a_i^t, \;\; \forall i \in \{1,\dots,T-1\}. \end{aligned} \tag{5.5}$$

$\bar{F}^*_F$ is a novel measure derived from the widely employed Forgetting metric [85] to facilitate the comparison between unevenly performing approaches. In particular, while the original Forgetting is upper-bounded by a model's accuracy, $\bar{F}^*_F$ varies in $[0, 100]$. $\bar{F}^*_F = 100$ denotes a method that

| **oTIL** | **Seq-CIFAR-100** | | **Seq-Mini-ImageNet** | |
|---|---|---|---|---|
| Joint-offline | 82.69 (–) | | 87.55 (–) | |
| Joint-online | 54.12 (–) | | 52.62 (–) | |
| Finetune | 35.42 (44.32) | | 31.55 (28.75) | |
| **Buffer Size** | 500 | 2000 | 2000 | 8000 |
| ER-ACE [40] | 56.06 (9.48) | 64.94 (3.19) | 64.68 (3.77) | 66.17 (4.10) |
| + CLER | **61.60**$^{JS}$ (9.21) | **69.33**$^{JS}$ (3.04) | **68.02**$^{R}$ (5.27) | **69.13**$^{JS}$ (4.11) |
| X-DER [86] | 63.10 (4.31) | 69.00 (1.38) | 67.67 (4.71) | 68.97 (4.39) |
| + CLER | **68.19**$^{JS}$ (2.98) | **73.45**$^{JS}$ (0.97) | **71.32**$^{JS}$ (3.01) | **72.39**$^{JS}$ (2.66) |
| CoPE [121] | 51.89 (23.46) | 66.56 (7.48) | 70.10 (4.89) | 73.61 (3.58) |
| + CLER | **60.19**$^{JS}$ (20.34) | **71.91**$^{JS}$ (6.42) | **71.17**$^{R}$ (5.30) | **75.33**$^{R}$ (2.54) |
| DualNet [43] | 49.38 (25.20) | 57.05 (13.85) | 68.43 (9.99) | 73.89 (5.54) |
| + CLER | **50.11**$^{R}$ (23.94) | **59.66**$^{JS}$ (12.99) | **70.26**$^{R}$ (7.39) | **76.97**$^{R}$ (3.87) |

***Table 5.2:*** *Final Average Accuracy (FAA) (↑) and Final Average Adjusted Forgetting ($\bar{F}^*_F$) (↓) on the oTIL setting.* $^{R}$ *indicates a result obtained with rotation,* $^{JS}$ *a result obtained with* $2 \times 2$ *jigsaw puzzle.*

retains no accuracy on previous tasks (*e.g.*, Finetune) and $\bar{F}^*_F = 0$ one that has no performance decrease on past tasks.

We repeat each experiment 10 times and report the mean FAA and $\bar{F}^*_F$, and the standard deviation of the former. Please refer to the supplementary material for the standard deviations and statistical significance.

### 5.4.2 Comparison with the State-Of-The-Art

We include the results of our evaluation on Seq-CIFAR-100 and Seq-Mini-ImageNet for oCIL and oTIL in Tab. 5.1 and 5.2 respectively. For each experiment, we report the best performer among the $2\times2$ jigsaw and rotation pretext tasks[2]. The evidence we present strongly supports our initial claims, with CLER improving the SOTA methods in all benchmarks. Specifically, we witness an improvement across the board regarding the FAA, while $\bar{F}^*_F$ indicates stronger resistance against forgetting.

Interestingly, the effect of our regularization is maintained regardless of the choice of buffer size, with an average oCIL improvement of $3.59$ and $3.40$ on Seq-CIFAR-100 and $3.12$ and $3.46$ on Seq-Mini-ImageNet. We find the only notable exception is in the case of DualNet on Seq-CIFAR-100. Indeed, even without our regularization, the lower FAA and higher forgetting compared with the other baselines suggests that the model cannot profit from the memory buffer. This might be due to the fact that the slow learner is only trained with a CSSL objective on samples from the buffer, which limits the quality of its representation when the latter is of moderate size. However, its results on the challenging Seq-Mini-ImageNet, when combined with CLER, suggest that such an effect can be mitigated by leveraging *equivariant* SSL, which allows the fast learner to develop better representations during OCL.

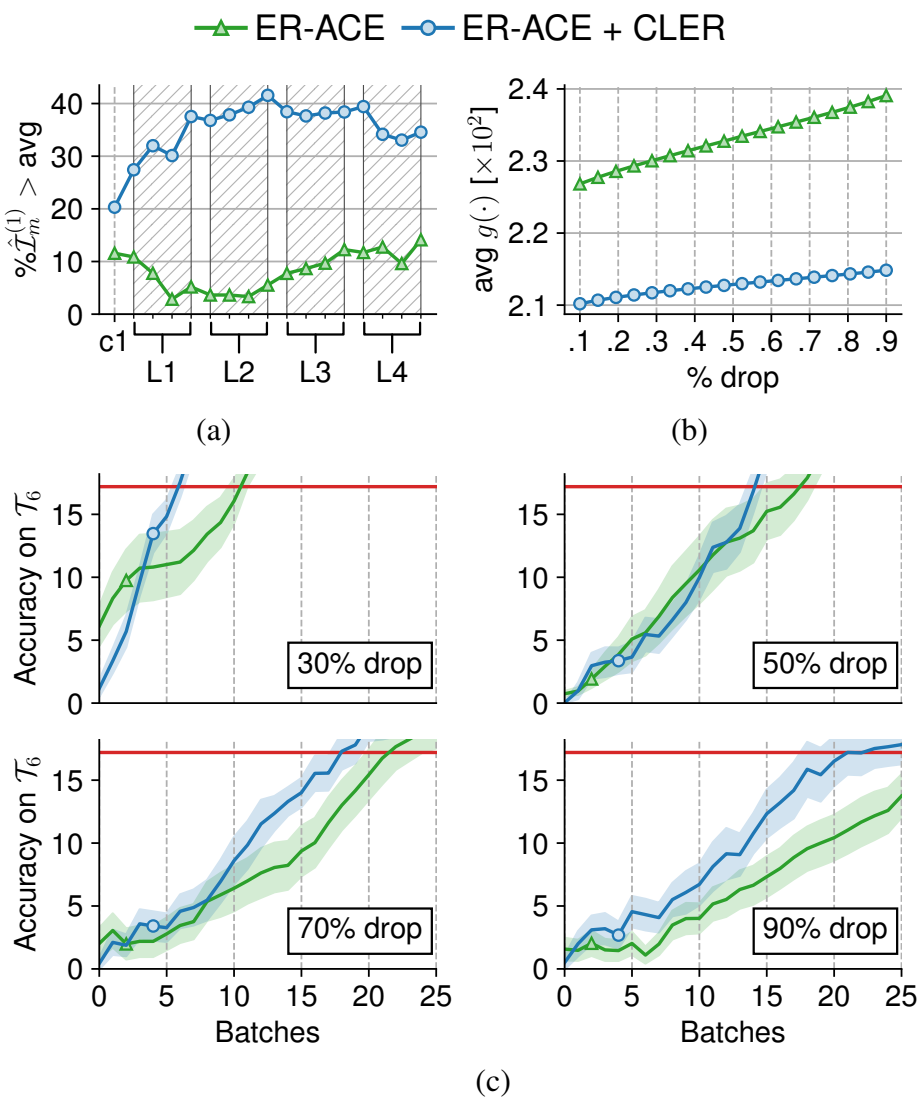



***Figure 5.3:*** *Structural analysis of ER-ACE with and without CLER on Seq-CIFAR-100. (a) Percentage of important neurons in each layer with higher-than-average importance score $\hat{\mathcal{I}}_m^{(1)}$; (b) within-layer similarity score $g$ after pruning with Geometric Median; (c) accuracy after dropping conv. filters and training on a few batches from $\mathcal{T}_6$, with the pre-drop accuracy serving as a target value (red line).*

## 5.5 Model Analysis

In the remainder, we analyze the various contributions of CLER and gather further insights on its overall effect on the CL tasks. To the best of our knowledge, our work is the first to consider the effect of equivariant-based pretext tasks in an incremental setting.

### 5.5.1 Effects of CLER on the Backbone

For an in-depth analysis of the effects induced on the backbone, we consider ER-ACE with and without CLER and conduct three additional experiments, drawing inspiration from the Network Pruning literature [132]. Our aim here is to unveil how the information carried by the learned features distributes across the parameters of the backbone.

#### Importance and redundancy

First, we quantify each parameter's contribution to the overall loss after training on Seq-CIFAR-100 by computing the *importance measure* $\hat{\mathcal{I}}_m^{(1)}$ proposed in [132]. In Fig. 5.3a, we focus on the convolutional layers and report the proportion of parameters whose importance score is higher than the layer's average to provide a compact per-layer evaluation.

Additionally, we perform a Geometric Median pruning [133] on the model, thus discarding those filters $\mathcal{F}_d$ that are the most redundant - *i.e.*, averagely most similar to all others in the same layer. In Fig. 5.3b we report

[2]Please refer to Sec. 5.5.2 for a detailed comparison between the two choices of pretext task.

| **Model** | **Seq-CIFAR-100 (oCIL)** | | **Seq-CIFAR-100 (oTIL)** | |
|---|---|---|---|---|
| **Buffer Size** | 500 | 2000 | 500 | 2000 |
| ER-ACE [40] | 20.17 (38.75) | 26.95 (23.69) | 56.06 (9.48) | 64.94 (3.19) |
| + CSSL | 20.89 (36.03) | 27.80 (21.12) | 56.22 (9.88) | 65.91 (2.42) |
| + CLER | **24.53**$^{JS}$ (33.76) | **30.89**$^{JS}$ (20.24) | **61.60**$^{JS}$ (9.21) | **69.33**$^{JS}$ (3.04) |
| X-DER [86] | 25.80 (39.54) | 30.44 (31.52) | 63.10 (4.31) | 69.00 (1.38) |
| + CSSL | 21.91 (36.07) | 23.59 (40.53) | 57.26 (2.76) | 62.56 (0.85) |
| + CLER | **29.35**$^{JS}$ (35.56) | **34.57**$^{JS}$ (29.71) | **68.19**$^{JS}$ (2.98) | **73.45**$^{JS}$ (0.97) |
| CoPE [121] | 19.98 (75.32) | 34.09 (46.39) | 51.89 (23.46) | 66.56 (7.48) |
| + CSSL | 17.23 (74.28) | 25.76 (54.72) | 49.56 (18.98) | 62.48 (3.64) |
| + CLER | **26.15**$^{JS}$ (69.28) | **38.48**$^{JS}$ (45.50) | **60.19**$^{JS}$ (20.34) | **71.91**$^{JS}$ (6.42) |

***Table 5.3:*** *Performance comparison between our proposal CLER and a similar Contrastive-based SSL (CSSL) method, as measured by Final Average Accuracy $FAA \pm \text{std}$ (↑) and Final Average Adjusted Forgetting ($\bar{F}^*_F$) (↓) on the Seq-CIFAR-100 benchmark.*

the average within-layer similarity $g$ for the discarded kernels:

$$g(\mathcal{F}_d) = \frac{1}{F}\sum_{j=1}^{F} |\mathcal{F}_d - \mathcal{F}_j|, \tag{5.6}$$

with $F$ the total number of filters in the considered layer.

Our results reveal that CLER pushes the model to fit the learned task with dense configurations of parameters (higher $\hat{\mathcal{I}}_m^{(1)}$ in Fig. 5.3a) that are also more similar to each other (lower $g$ in Fig. 5.3b). We conjecture that this can be linked to the performance increase reported in Sec. 5.4.2: as the knowledge of a specific task does not rely on only a few parameters but

instead appears more distributed, it is less likely that subsequent weights' updates will entirely erase the previously acquired knowledge. Moreover, the higher rate of important parameters, coupled with the higher redundancy, suggests that those important filters erased by forgetting could be restored as needed, by simply leveraging redundant groups of parameters.

#### Recovery

To support our intuitions, we conducted an additional evaluation probing the dynamics of learning with CLER. After training on the $6^{\text{th}}$ task of Seq-CIFAR-100, we randomly drop a portion of the convolutional filters in our models and retrain using only the cross-entropy loss on a few batches from the same task, reporting the accuracy after each batch in Fig. 5.3c. Interestingly, the distributed importance induced by our training objective leads to a higher initial drop in accuracy for CLER. However, our proposed approach swiftly recovers its performance, reaching the target pre-drop accuracy in fewer steps w.r.t. the baseline.

### 5.5.2 Invariance & Equivariance

While in previous sections we explored the role of equivariance as a regularizer for OCL, we now wish to better characterize the different pretext tasks, as well as compare with an invariance-based CSSL objective.

#### Rotations *vs* Jigsaw

The results presented so far depict a clear advantage of the jigsaw puzzle pretext task, which might suggest that the performance gain is not specifically tied to equivariance but to the former. To address such concern, in

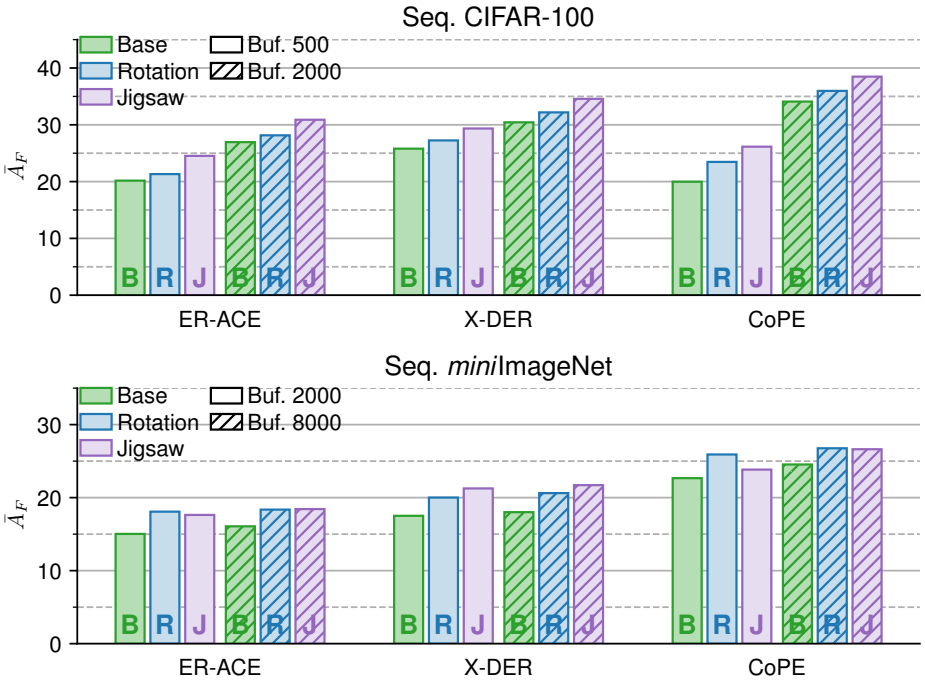



***Figure 5.4:*** *Final Average Accuracy (FAA) [↑] of various baseline methods when equipped with different equivariant pretext tasks: four-fold rotation prediction and* $2 \times 2$ *jigsaw solving. Both methods achieve higher results w.r.t. the baseline, with jigsaw solving usually leading to the best performance.*

Fig. 5.4 we present detailed results for the evaluation of Sec. 5.4.2 on the oCIL setting both with four-fold rotation and jigsaw puzzle. Our results depict a clear advantage of both equivariant pretext tasks w.r.t. the baseline method. Moreover, the similar performance achieved by the two (especially on the challenging Seq-Mini-ImageNet benchmark) further proves our initial assumption about the effectiveness of equivariant-based SSL methods in CL.

### Comparison with CSSL methods

Our initial analysis shows that enforcing *equivariance* to a set of input transformations efficiently allows CLER to learn a representation robust against forgetting, by spreading the contribution of each feature on all the learnable parameters. This is in contrast with current CL literature, which instead relies on CSSL tasks [41, 43] to learn a representation that is *invariant* to strong data augmentation and input transformations.

To further prove our contribution, in Tab. 5.3 we compare our proposal of an equivariant loss term against one that promotes invariance by means of a CSSL objective. For the latter, we take inspiration from [43] and opt for Barlow Twins. Our results indicate a superior regularization effect for CLER, with CSSL even hurting the performance in some scenarios. This suggests that the few training iterations allowed in OCL do not allow CSSL to transfer useful knowledge, thus eventually hindering incremental learning.

### Applicability to the multi-epoch setting

While we focus our evaluation on OCL, we reckon that our proposed approach might also prove beneficial in a less strict environment that allows

| | ER-ACE [40] | + CSSL | + CLER |
|---|---|---|---|
| **Epochs** | **Buffer size** 500 | | |
| 1 (OCL) | 20.17 (38.75) | 20.89 (36.03) | **25.08**$^{JS}$ (32.84) |
| 5 | 32.47 (47.70) | 33.53 (46.29) | **34.88**$^{JS}$ (45.52) |
| 20 | 37.38 (46.79) | 37.78 (50.55) | **39.35**$^{JS}$ (46.84) |
| 50 | 37.94 (51.49) | 39.61 (43.75) | **41.27**$^{JS}$ (46.78) |
| **Epochs** | **Buffer size** 2000 | | |
| 1 (OCL) | 26.95 (23.69) | 27.80 (21.12) | **30.89**$^{JS}$ (20.24) |
| 5 | 42.35 (27.49) | 43.62 (27.11) | **45.67**$^{JS}$ (24.92) |
| 20 | 48.03 (33.33) | 49.16 (31.86) | **50.27**$^{JS}$ (31.20) |
| 50 | 49.05 (33.91) | 50.66 (34.48) | **52.17**$^{JS}$ (32.56) |

***Table 5.4:*** *Performance comparison for Equivariant- and Contrastive-based SSL objectives in a multi-epoch setting, evaluated on Seq-CIFAR-100. We measure the Final Average Accuracy (FAA) [↑] and find generally stronger performance for CLER even when the online constraint is relaxed.*

for multiple iterations. Such a setting simulates a realistic low-latency scenario, where the desiderata is an algorithm capable of rapidly adapting to the changing data stream while retaining knowledge from the past. Results of this evaluation on the Seq-CIFAR-100 benchmark are summarized in Tab. 5.4. Due to space constraints, we only include results on the Class-Incremental scenario.

Unsurprisingly, as the number of epochs increases, the model can start to fully leverage the knowledge that comes from the stream. However, as CSSL tasks usually require a large number of iterations to converge, our

| Method | Seq-CIFAR-100 | Seq-Mini-ImageNet |
|---|---|---|
| **Joint-offline** | 69.85±1.43 | 62.42±1.13 |
| + CSSL | 70.24±0.47 | 63.10±0.61 |
| + CLER | 70.92[JS]±0.74 | 63.11[JS]±0.16 |
| **Joint-online** | 23.14±0.74 | 10.68±0.67 |
| + CSSL | 23.16±0.82 | 13.79±0.79 |
| + CLER | 28.38[JS]±1.82 | 14.77[JS]±0.78 |

***Table 5.5:*** *Accuracy of Joint methods with CSSL and CLER. The epochs are set to 30, 50 for Seq-CIFAR-100 and Seq-Mini-ImageNet respectively.*

CLER remains a better choice for the task of preventing forgetting while boosting the representation of the base model.

### 5.5.3 Is CLER's advantage actually tied to OCL?

The consistently enhanced performance of baseline methods when combined with CLER could raise the suspicion that SSL regularization is generally effective and not particularly relevant to Continual Learning *per se*. To shed light on this point, we apply both CSSL and CLER regularization on a multi-epoch Joint upper bound (Joint-offline) and report the results in Tab. 5.5; this simple test clearly shows that – if enough epochs are allowed and the method achieves full convergence – the presence of additional SSL terms does not impact the attained accuracy significantly.

To complement this result, we also apply the proposed technique on top of single-epoch Joint training. In this context, CLER proves effective and more so than CSSL. In line with what shown in Fig. 5.1, this result

| Method | Seq-CIFAR-100 | Seq-Mini-ImageNet |
|---|---|---|
| LWF.MC [31] | 36.15 (49.78) | 20.75 (63.67) |
| + CLER | **37.07**$^{R}$ (49.37) | **21.64**$^{R}$ (62.79) |
| R-DFCIL [134] | 34.98 (54.59) | 13.15 (83.47) |
| + CLER | **36.74**$^{R}$ (52.31) | **18.80**$^{JS}$ (75.43) |

***Table 5.6:*** *Class-IL Final Average Accuracy (FAA) [↑] of DFCIL methods (no buffer) with and without CLER. We conduct 30, 50 epochs on Seq-CIFAR-100, seq-Mini-ImageNet respectively.*

confirms that SSL facilitates the convergence of the learner when having only few data-points and that the equivariant approach of CLER is more sample-efficient than typical CSSL methods.

In conclusion, we summarize that **self-supervised regularization is not effective in a multi-epoch non-continual setting** (Tab. 5.5 *top*); it becomes relevant in either single-epoch (Tab. 5.5 *bottom*) or continual (Tab. 5.4) setting. Due to its enhanced sample efficiency, **the equivariant approach pursued by CLER is particularly effective when fewer epochs are performed**. For this reason, its application is ideal for the OCL setting.

### 5.5.4 Applicability to Data-Free Continual Learning

The SOTA competitors on top of which we validate CLER in Sec. 5.4 belong to the rehearsal-based family of CL methods. These represent by far the preferred approach in the challenging oCIL scenario, on which the performance of other classes of methods is severely compromised [120, 40, 135, 136]. However, a very recent line of works raises criticism on the

adoption of replay, citing potential privacy issues [137, 134]. They instead focus on the so-called **Data-Free Class-Incremental Learning (DFCIL)** setting, *i.e.*, **multi-epoch** Class-Incremental Learning without a memory buffer.

To provide a clear picture of the flexibility of our proposal, we further showcase its application on top of two DFCIL methods: the model inversion-based Relation-Guided Representation Learning (R-DFCIL) [134] and the distillation-based Multi-Class Learning without Forgetting (LWF.MC) [31]. The results in Tab. 5.6 illustrate that CLER delivers a steady performance improvement even in DFCIL, which reveals that its effectiveness is not dependent on the availability of replay data.

## 5.6 Discussion

We present **Continual Learning via Equivariant Regularization** (**CLER**), a novel approach for *Online Continual Learning* (OCL) that encourages representations to be sensitive to a set of input transformations. Our method introduces a regularization technique based on equivariant SSL pretext tasks (jigsaw puzzle solving and four-fold rotation prediction). By experimental means, we show that the application of CLER to state-of-the-art methods consistently leads to better performance. Furthermore, we provide an in-depth analysis of the effect of CLER on the parameters of the backbone network and compare it against other Contrastive Self-Supervised Learning methods.

Our strong results with different choices of equivariant pretext tasks further support our initial hypothesis, laying the foundation for better OCL models based on equivariant constraints. We leave this analysis for future

work.

## 5.7 Publications

The approach described in this chapter is currently under review at IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) journal.

Bonicelli, L., Boschini, M., Frascaroli, E., Porrello, A., Pennisi, M., Bellitto, G., Palazzo, S. Spampinato, C., Calderara, S. (2023). On the Effectiveness of Equivariant Regularization for Robust Online Continual Learning. Submitted to IEEE Transaction on Pattern Analysis and Machine Learning.

# Part III

# TOWARDS NEUROCOGNITIVE CONTINUAL LEARNING

*"Dreams feel real while we're in them. It's only when we wake up that we realize something was actually strange."*

Dom Cobb, *Inception*, 2010

The intricate and adaptable structure of human neural connections allows for the seamless assimilation of knowledge across diverse contexts. This adaptability underscores a stark contrast between human cognition and the current capabilities of AI systems. While both entities process information, there are striking differences in their perceptual systems, neural architectures, and learning paradigms. Compounding these differences is our limited understanding of the fundamental mechanisms of the human brain, which poses a significant challenge to replicating its capabilities.

To bridge the gap between machine and human cognition, the second part of the thesis is devoted to methods that draw inspiration from neurocognitive theories. These theories, while still experimental, provide invaluable insights into human cognitive processes. Two processes stand out: first, the inherent visual attention mechanisms that have evolved in humans over millennia; and second, the critical role of off-line states, particularly sleep, in consolidating memories and forming new semantic structures.

In Chapter 6, the focus shifts to the human visual system, a sophisticated network of structures fundamental to the processing of visual stimuli from the environment. Taking advantage of the specific neurophysiological features of the primary visual cortex, we propose a method that uses auxiliary saliency prediction features to improve the stability and accuracy of learning sequences in non-i.i.d. classification tasks.

Chapter 7 explores the enigmatic but essential domain of dreaming and its role in cognitive functions. Far from being a merely passive state, sleep actively facilitates the reinforcement and integration of neural representations of new experiences into established knowledge matrices. Drawing on the dualistic nature of wake-sleep memory acquisition in humans, this chapter explores the potential of these mechanisms to inform and innovate knowledge storage strategies in neural networks.

CHAPTER

# SIX

# SELECTIVE ATTENTION-BASED MODULATION FOR CONTINUAL LEANING

In Chapter 5, we explored the effective combination of a continuously trained classifier combined with an additional task to guide the learning process. In this chapter, we continue this perspective by taking an approach inspired by how the human brain works, in particular by drawing insights from some peculiarities of the Human Visual System (HVS) [138, 139]. The HVS is a complex and sophisticated networks of structures and processes responsible for processing visual information from the environment. It includes the eyes, and various components of the central nervous system (retina, optic nerve, optic tract and visual cortex). It empowers humans to see, perceive and comprehend the world around them.

As we delve into bio-inspired methodologies, we uncover valuable insights that pave the way for enhancing classification models in a continual learning setting. Inspired by neurophysiological evidence that the primary

visual cortex does not contribute to object manifold untangling for categorization and that primordial attention biases are still embedding in the modern brain, here we propose to employ auxiliary saliency prediction features as a modulation signal to drive and stabilize the learning of a sequence of non-i.i.d. classification tasks.

## 6.1 Motivation

Humans possess the remarkable capability to keep learning, with limited forgetting of past experience, and to quickly re-adapt to new tasks and problems without disrupting consolidated knowledge. Machine learning, on the contrary, has shown significant limitations when dealing with non-stationary data streams with a limited possibility to replay past examples. The main reason for this shortcoming can be found in the inherent structure, organization and optimization approaches of artificial neural networks, which differ significantly from how humans learn and how their neural connectivity is built when accumulating knowledge over a lifetime. According to the *Complementary Learning Systems (CLS) theory* [140, 141], the human ability to learn effectively may be due to the interplay between two learning processes that originate, respectively, on the hippocampus and on the neocortex. These two brain regions interact to support learning representations from experience (the neocortex) while consolidating and sustaining long-term memory (the hippocampus). This theory has inspired several continual learning methods [142, 143, 144]. In particular, the recent DualNet method [43] translates CLS concepts into a computational framework for continual learning. Specifically, it employs two learning networks: a *slow learner*, emulating the memory consoli-

dation process happening in the hippocampus through contrastive learning techniques, and a *fast learner*, that aims at adapting current representations to new observations.

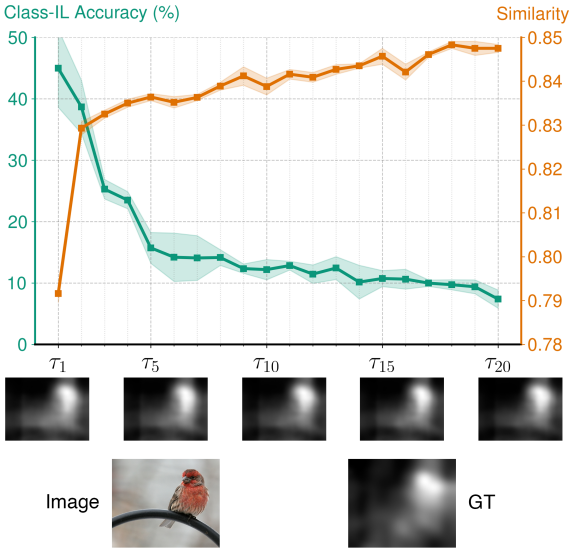



***Figure 6.1: Comparison between the forgetting-free behavior of saliency prediction and the typical catastrophic forgetting observed on classification tasks in continual learning scenarios.*** *Saliency accuracy (measured as* similarity *[145]) improves as the saliency network is presented with more tasks, while classification accuracy drops. This suggests that saliency detection is an i.i.d. task even in presence of a non-i.i.d. data distribution. Images on the $x$ axis show how predicted saliency maps are approximately constant over tasks.*

However, this strategy still appears insufficient for addressing the problem of continual learning, because it starts from the (possibly wrong) assumption that human neural networks directly process visual input with

the objective of performing categorization from early vision layers. On the contrary, neurophysiological studies [146, 147] are in near universal agreement that the object manifolds conveyed to primary visual cortex V1 (one of the earliest areas involved in vision) are as tangled as the pixel space. In other words, the neurons of the earliest vision areas do not contribute to object manifold untangling for categorization, but rather enforce luminance and contrast robustness [147]. This suggests that training early neurons with a visual categorization objective — as done not only in DualNet, but in all existing continual learning methods — is in stark contrast to the biological counterparts observed in primates. Moreover, recent studies on the causes of forgetting in artificial neural networks showed that deeper layers (i.e., closer to the output) are less stable in presence of task shifts [148], which is consistent with the hypothesis that earlier layers do not bear specific categorization responsibilities.

Given these premises, it is peculiar that existing bio-inspired continual learning methods tend to ignore all upstream neural processes underlying visual categorization, such as visual attention. Indeed, the ability to select relevant visual information appears to be the hallmark of human/primate cognition. Moreover, recent findings in cognitive neuroscience have shown that the visual attention priorities of human hunter-gatherer ancestors are still embedded in the modern brain [149]: humans pay attention faster to animals than to vehicles, although we now see more vehicles than animals. This primordial attention bias embedded in human brains suggests that the neuronal circuits of the ventral visual pathway are somehow inherited, as a form of genetic legacy from ancestral experience, and tend to remain stable over time — thus not subject to forgetting, though we have long stopped hunting to survive.

Interestingly, we observed the same **forgetting-free** behavior for

saliency prediction on artificial neural networks. Fig. 6.1 shows the trend of the *similarity* [145] metric for a saliency prediction model trained in a continual learning scenario, and compares it to the accuracy of a classification model under the same settings. While classification accuracy drops as the classifier learns new classes, the saliency metric remains stable, and even slightly improves.

From this observation, in this paper we propose *SAM*, a *Selective Attention-driven Modulation* strategy that employs saliency prediction [150] to drive the learning of a sequence of classification tasks in a continual learning setting. To emulate what has been observed in primates, where visual attention modulates the firing rate of neurons that represent the attended stimulus at different stages of visual processing [151, 152], SAM adopts a two-branch model: one branch performs visual saliency prediction [153, 154, 155], and its responses modulate (through multiplication) the features learned by a paired classification model in the second branch. SAM is model-agnostic and can be used in combination to any continual learning method. We demonstrate that saliency modulation positively impacts classification performance in online continual learning settings, leading to a significant gain in accuracy (up to 20 percent points) w.r.t. baseline methods. We further demonstrate the usefulness of saliency modulation on different benchmarks (including a challenging one that tackles fine-grained classification) and substantiate our claims through a set of ablation studies. We finally show that saliency modulation, besides being biologically plausible, leads to learn saliency-modulated features that are more robust to the presence of spurious features and to adversarial attacks.

In summary, we make the following contributions:

- We introduce a new continual learning strategy named **Selective Attention-driven Modulation** (SAM), where the image classifier is coupled with a saliency prediction model that drives its learning by effectively reducing forgetting. Interestingly, our approach is model-agnostic and can be easily used with all existing methods.

- We evaluate our SAM strategy in the more complex Online Continual Learning scenario, on the well-established Seq-Mini-ImageNet [47, 130, 131] benchmark, and on a even more challenging dataset containing only image of animals. We discover that methods trained according to our strategy significantly increase the final accuracy, outperforming existing multi-branch solutions like DualNet [43], CoPE [121] and TwF [156].

- We show that saliency modulation, besides being biologically plausible, leads to learn saliency-modulated features that are more robust to the presence of spurious features and to adversarial attacks.

## 6.2 Related Work

### Continual Learning

Continual Learning (CL) [18, 46] is a recently-popularized branch of machine learning whose objective is to bridge the gap in incremental learning between humans and neural networks. McCloskey and Cohen [5] highlight that the latter experience a *catastrophic forgetting* of previously acquired knowledge in the presence of distribution shifts in the input data stream. To compensate for this problem, countless solutions have been proposed that

introduce either adequate regularization terms [23, 25], specific architectural organization [24, 27] or the rehearsal of a small number of previously encountered data points [32, 31, 38].

While current solutions help mitigating forgetting, their application to real-world settings proves difficult, as typical CL evaluations are conducted in accordance to unrealistic benchmarks [157, 158]. *Online CL* (OCL) [120] addresses this issue by forbidding multiple epochs on the input stream. This is meant to model the realistic assumption that any data point captured in the wild occurs only once.

To reach reasonable performance, most approaches tackling this challenging scenario adopt a replay strategy [6, 32]. Some works focus on memory management: GSS [34] introduces a specific optimization of the basic rehearsal formula meant to store maximally informative samples in memory, while HAL [126] individuates synthetic replay data points that are maximally affected by forgetting. Other approaches propose tailored classification schemes: CoPE [121] uses class prototypes to ensure a gradual evolution of the shared latent space; ER-ACE [40] makes the cross-entropy loss asymmetric to minimize imbalance between current and past tasks. Finally, other works introduce a surrogate optimization objective: SCR [159] employs a supervised contrastive learning objective and OCM [160] leverages mutual information objectives: both aim at learning informative features that are less subject to forgetting.

Our proposal adopts a remarkably different approach w.r.t. these classes of methods, in that we take inspiration from cognitive neuroscience theory of learning and exploiting the features of a conjugate forgetting-free task (i.e., saliency prediction) to modulate the responses of our OCL model. Doing so produces a stabilizing effect on our model and makes it more resilient to forgetting.

An approach that is similarly inspired by cognitive theories is DualNet [43], which employs two networks that loosely emulate how slow and fast learning work in humans. However, DualNet employs contrastive learning on the slow network (the earliest layers of the model), while it seems that object-identifying transformations happens later in the human visual system [146, 147]. Our results, reported later, substantiate the suitability of our choice to use low-level processes, such as selective attention, to drive continual learning tasks, rather than contrastive learning or classification pre-training techniques as, respectively, in DualNet and TwF [156]. Finally, our work, like DualNet, follows the emerging *NeuroAI* [161] paradigm promoted by deep learning pioneers, according to which human neural computation will drive the next revolution in AI, bringing machines closer to human capabilities.

Despite the idea of using saliency prediction maps in continual learning has never been proposed, we have assisted to a recent trend where forgetting can be mitigated if the model is encouraged to recall the evidence for previously made decisions, stored as activation maps [162]. Specifically, it employs explainability techniques as Gradient-weighted Class Activation Mapping (Grad-CAM [163]) to store visual model explanations for each sample in the buffer and ensures model consistency with previous decisions during the training phase. Similarly, EPR [164], instead of retaining whole images, employs Grad-CAM to identify the important patches and stores them in the episodic memory.

The above methods rely on using activation maps (sometime referred to as saliency maps) as regularizers to limit forgetting. However, it is worth to highlight that there exist a fundamental difference with our approach. Within the context of explainable artificial intelligence (XAI), techniques as Grad-CAM are utilized to generate *attribution maps* to support a model

prediction in term of relevant input features. While they aim to identify important visual areas for a pre-trained classifier, a saliency predictor is a neural network trained with the aim to predict the area of a scene that will capture the attention of a human observer. Moreover, as reported in our experimental results, attribution maps tend to degrade over time, since they strictly depend on the internal state of a neural network, which is subject to forgetting. In contrast, our saliency-based modulation, stemming directly from neuroscience theory and from our finding of forgetting-free behaviour of saliency is novel and unexplored.

### Saliency Prediction

When exposed to visual stimuli, being either *static* (image) or *dynamic* (video) scenes, humans have the ability to focus visual attention toward the area of the scene that contain the most important information. Saliency prediction is the task of predicting the gaze fixation of an observer when viewing a scene, and it has long been investigated by computer vision researchers. Initially, prior works focused exclusively on images. Static saliency was studied extensively, using biological-inspired methods [165] and employing hand-crafted features [166, 167, 168]. They are also referred as *bottom-up* methods, since they focused on low-level features, such as contrast, colors, edge, etc. Thereafter, with the emergence of deep learning and CNNs, a new plethora of methods, termed *top-down*, have been proposed, achieving superior results and establishing the new state-of-the-art [169, 170, 171, 172, 173, 174, 155]. Most of them are based on the idea of exploiting existing pre-trained image classification model (such as VGG, ResNet or DenseNet) as saliency encoders, while several decoder architectures have been introduced.

Dynamic saliency is more complex than static saliency because it involves the temporal dimension and requires additional computational complexity, but it is arguably more relevant to human visual experience. Early attempts were based on adapting methods originally designed for image saliency and applying them frame-by-frame. To extract temporal information, prior video saliency models relied on optical flow [175] or LSTM modules [176, 177, 178]. These methods outperforms state-of-the-art image saliency model because the latter do not use any temporal or motion information. Recently, the release of a new large-scale dataset for video saliency prediction, i.e. the DHF1K benchmark [178], has allowed newer methods to take a step further in performance. The latest TASEDNet [179], HD$^2$S [180] and STSANet [181] exploit 3D Convolutional-based encoders that, unlike previous methods, allow spatial and temporal information to be jointly processed, while VSFT [182] and TMFI [183] adopt a spatio-temporal encoder based on Transformers [184] architecture, setting the current state-of-the art.

## 6.3 Method

### 6.3.1 Online Continual Learning

Following the recent literature, we pose OCL as a supervised image classification problem with an online non-i.i.d. stream of data, where each training sample is only seen once. Although our attention-driven modulation does not require the presence or knowledge of *task boundaries*, in this formulation and in our experiments we assume that these are given, to the benefit of any baseline method enhanced by the proposed extension.

More formally, let $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$ be a sequence of data streams,

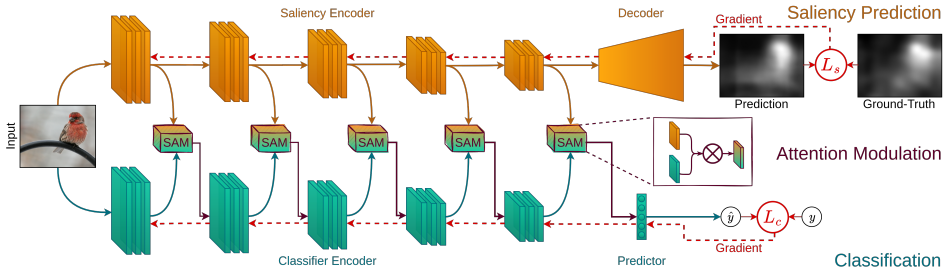



***Figure 6.2: Architecture of the proposed selective attention-based modulation (SAM) strategy.*** *The classification backbone is paired with a saliency prediction network that, given its capability of being forgetting-free, aims at adjusting the learned classification features in order to mitigate overall forgetting.*

where each pair $(\mathbf{x}, y) \sim \mathcal{D}_i$ denotes a data point $\mathbf{x} \in \mathcal{X}$ with the corresponding class label $y \in \mathcal{Y}$; the sample distributions (in terms of both the data point distribution and the class label distribution) of different $\mathcal{D}_i$ and $\mathcal{D}_j$ may vary — for instance, class labels from $\mathcal{D}_i$ might be different from those from $\mathcal{D}_j$, though both must belong to the same domain $\mathcal{Y}$. Given a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$, parameterized by $\boldsymbol{\theta}$, the objective of OCL is to train $f$ on $\mathcal{D}$, organized as a sequence of $T$ tasks $\{\tau_1, \ldots, \tau_T\}$, under the constraint that, at a generic task $\tau_i$, the model receives inputs sampled from the corresponding data distribution, i.e., $(\mathbf{x}, y) \sim D_i$, and sees each sample only once during the whole training procedure. The classification model may optionally keep a limited *memory buffer* $\mathbf{M}$ of past samples, to reduce forgetting of features from previous tasks. The model update step between tasks can be summarized as:

$$\langle f, \boldsymbol{\theta}_{i-1}, \mathcal{D}_{i-1}, \mathbf{M}_{i-1} \rangle \rightarrow \langle f, \boldsymbol{\theta}_i, \mathbf{M}_i \rangle \tag{6.1}$$

where $\boldsymbol{\theta}_i$ and $\mathbf{M}_i$ represent the set of model parameters and the buffer at the end of task $\tau_i$, respectively. For methods that do not exploit buffer, $\mathbf{M}_i = \emptyset, \forall i$.

The training objective is to optimize a classification loss over the sequence of tasks (without losing accuracy on past tasks) by the model instance at the end of training:

$$\arg\min_{\boldsymbol{\theta}_T} \sum_{i=1}^{T} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i} \Big[ \mathcal{L}\Big( f\left(\mathbf{x}; \boldsymbol{\theta}_T\right), y \Big) \Big] \tag{6.2}$$

where $\mathcal{L}$ is a generic classification loss (e.g., cross-entropy), which a continual learning model attempts to optimize while accounting for model *plasticity* (the capability to learn current task data) and *stability* (the capability to retain knowledge of previous tasks) [5].

### 6.3.2 SAM: Selective Attention-driven Modulation

Our method is grounded on the neurophysiological evidence that attention-driven neuronal firing rate modulation is multiplicative and the scaling of neuronal responses depends on the similarity between a neuron's preferred stimulus and the attended feature [151, 152]. This hypothesis is translated into a general artificial neural architecture, where we emulate the the process of human selective attention through a visual saliency prediction network whose activations modulate, through multiplication, neuron activations of a paired classification network at different stages of visual processing. Formally, let $S : \mathcal{X} \rightarrow \mathcal{S}$ be a saliency prediction network, where $\mathcal{X}$ is the space of input images and $\mathcal{S}$ the space of output saliency maps. Generally, if $\mathcal{X} = \mathbb{R}^{3\times H\times W}$ for RGB images, then $\mathcal{S} = \mathbb{R}^{H\times W}$,

where each location of a map $\mathbf{s} \in \mathcal{S}$ measures the *saliency* of the corresponding pixel in the RGB space. We assume that $S$ can be decomposed into two functions, an encoder $E : \mathcal{X} \rightarrow \mathcal{H}$ and a decoder $D : \mathcal{H} \rightarrow \mathcal{S}$, such that $S(\mathbf{x}) = D(E(\mathbf{x}))$, for $\mathbf{x} \in \mathcal{X}$. Then, given an online continual learning problem with data stream $\mathcal{D}$ and set of classes $\mathcal{Y}$, let $C : \mathcal{X} \rightarrow \mathcal{Y}$ be a classification network, such that $C$ and the saliency encoder $E$ share the same architecture (with independent parameters). An illustration of the proposed architecture is shown in Fig. 6.2.

At training time, both $S$ and $C$ observe the same data stream, from which pairs $(\mathbf{x}, y)$ of input data and class label are iteratively sampled. Through the use of an external *saliency oracle*, we extend each data sample to a triple $(\mathbf{x}, y, \mathbf{s})$, where $\mathbf{s}$ is the target saliency map associated to $\mathbf{x}$. The oracle can be either a set of ground-truth maps, when available, or *pseudo-labels* provided as the output of a pre-trained saliency predictor (unrelated to $S$). We therefore proceed to optimize a multi-objective loss function $\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_c$, with $\lambda$ being a weighing hyperparameter. Loss term $\mathcal{L}_s$ is computed on the output of saliency predictor $S$, and compares the estimated saliency map $S(\mathbf{x})$ with the target $\mathbf{s}$ by means of the Kullback-Leibler divergence (commonly employed as a saliency prediction objective [145, 155, 180, 181, 185]):

$$\mathcal{L}_s = \sum_i s_i \log \left( \frac{s_i}{S_i(\mathbf{x}) + \epsilon} + \epsilon \right) \tag{6.3}$$

with $s_i$ and $S_i(\mathbf{x})$ iterating over map pixels in $\mathbf{s}$ and $S(\mathbf{x})$, respectively. Loss term $\mathcal{L}_c$ encodes a generic online continual learning objective, as introduced in Eq. 7.2. As the proposed approach is method-agnostic, details on the formulation of $\mathcal{L}_c$ may vary.

In order to enforce selective attention-driven modulation of classifica-

tion neuronal activations, we leverage the architectural identity of saliency prediction encoder $E$ and classifier $C$ to alter the feedforward pass of the latter, by multiplying pre-activation features in $C$ by the corresponding features in $E$, before applying a non-linearity and feeding them to the next layer of the network. Formally, let us assume that the $C$ and $E$ networks consist of a sequence of layers $\{l_1, l_2, \ldots, l_L\}$. Without loss of generality, let each layer $l_i$ compute its output as $\mathbf{z}_i = \sigma\left(\mathbf{W}_i \mathbf{z}_{i-1}\right)$, with $\sigma$ being an activation function, $\mathbf{W}_i$ the network-specific layer parameters (i.e., not shared between $E$ and $C$) and $\mathbf{z}_{i-1}$ the output of the previous layer (or the network's input $\mathbf{x}$, if appropriate). Then, let us distinguish between features $\mathbf{z}_i^{(s)}$ and $\mathbf{z}_i^{(c)}$, respectively representing the output of layer $l_i$ by the saliency prediction encoder $S$ and the classifier $C$. We apply attention-driven modulation by modifying the computation of $\mathbf{z}_i^{(c)}$ as follows:

$$\mathbf{z}_i^{(c)} = \sigma\left(\mathbf{W}_i^{(c)}\left(\mathbf{z}_{i-1}^{(c)} \odot \mathbf{z}_{i-1}^{(s)}\right)\right) \tag{6.4}$$

where $\odot$ denotes the Hadamard product. Intuitively, the proposed approach encourages the classification model to attend to "salient" features of the input, where the concept of *saliency* is generalized from the pixel space to hidden representations. It is important to note that, at training time, gradient descent optimization of $\mathcal{L}_c$ would also affect on the saliency encoder $E$. This is undesirable, as we previously showed (see Fig. 6.1) that saliency features are robust to task shifts, unlike classification features: hence, in order to guarantee this property, we stop the gradient flow from $\mathcal{L}_c$ to parameters in $E$, and use it to update the parameters of classifier $C$ only.

In the above formulation, we assumed the presence of a classification network with fully-connected layers; however, our method can be applied

in an agnostic manner to any method employing, at least in part, a feature extractor implemented as a neural network. As such, the proposed method can be equally applied, for instance, both to end-to-end classification models (e.g., DER++ [38]) and to approaches with a neural backbone that computes class-representative prototypes (e.g., CoPE [121]).

## 6.4 Experimental Results

### 6.4.1 Benchmarks

We build two OCL benchmarks by taking image classification datasets and splitting their classes equally into a series of disjoint tasks:

- **Seq-Mini-ImageNet** [47, 130, 131] is a challenging dataset that includes 100 classes from ImageNet, allowing for a longer task sequence. For each class, 500 images are used for training and 100 for evaluation.
- **Seq-FG-ImageNet**[1] is a benchmark for fine-grained image classification that we use to test CL methods on a more challenging task than traditional ones. It includes 100 classes of animals extracted from ImageNet, belonging to 7 different species (*annelids*, *arachnids*, *birds*, *clams*, *fishes*, *reptiles*, *shellfish*), reducing inter-class variability and leading to harder tasks. Each class contains 500 samples for training and 50 for evaluation.

For both datasets, images are resized to 288×384 pixels and split into twenty 5-way classification tasks.

[1] Seq-FG-ImageNet is derived from https://www.kaggle.com/datasets/ambityga/imagenet100

### 6.4.2 Training and Evaluation Procedure

#### Baseline Methods

We evaluate the contribution of the SAM strategy when paired to a classification network trained using several state-of-the-art continual learning approaches, including rehearsal and non-rehearsal methods:

- **DER++** [86]: a seminal work that combines rehearsal and knowledge distillation strategies for supporting model plasticity while limiting forgetting.
- **ER-ACE** [40]: a variant of experience replay [6, 32] which aims to prevent imbalances due to the simultaneous optimization of the current and past tasks by selectively masking softmax outputs.
- **CoPE** [121]: a prototype-based classifier with experience replay, whose careful update scheme prevents sudden disruptions in the latent space during incremental learning.
- **LwF** [21]: a non-rehearsal method that enforces a model to preserve outputs of past model instances on new samples to limit forgetting.
- **oEWC** [186]: a non-rehearsal method that mitigates forgetting by selectively limiting the changes on weights that are most informative of past tasks.

All above methods employ ResNet-18 [69] as a feature extraction backbone. We also report the results of jointly training the model on all classes for one epoch (**Joint**), and of training sequentially on each task without any particular countermeasure for avoiding forgetting (**Fine-tune**).

### Implementation details

We apply the SAM strategy at five feature modulation points of ResNet-18's architecture, namely, the outputs of the first convolutional block and of the four main residual blocks. In compliance with online learning, all models are trained for a single epoch, using SGD as optimizer, with a fixed batch size of 8 both for the input stream and the replay buffer. Rehearsal methods are evaluated with three different sizes of the memory buffer (1000, 2000 and 5000). When applying SAM, besides each method's specific training objective, we also optimize the saliency prediction loss $\mathcal{L}_s$ from Eq. 6.3. Saliency is estimated using DeepGaze IIE network [153] as oracle.

When using SAM, classifier $C$ and saliency predictor $S$ are identical ResNet-18 architectures, followed — respectively — by a linear classification layer and a saliency map decoder. While $C$ is trained from scratch, we employ a pre-trained saliency predictor $S$, consistently with neuroscience evidence showing that humans have selective attention already embedded in the brain [149]. For a fair comparison, feature extraction backbones of baseline methods are initialized to the same pre-trained weights as $S$. Care was taken to ensure that the set of OCL classes $\mathcal{C}$ did not semantically overlap with pre-training data, to prevent any contamination from the saliency predictor to the classification task. Specifically, $S$ was pre-trained for 20 epochs on a subset of 100 ImageNet classes (disjoint from our two main benchmark datasets), using DeepGaze IIE as oracle. No class label information was used at this stage.

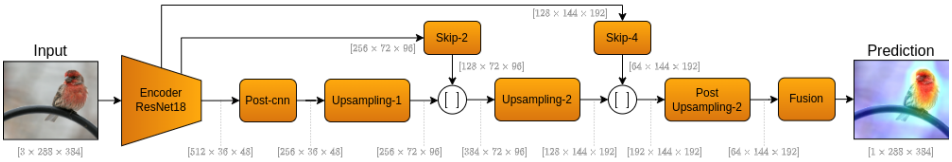



***Figure 6.3: Overview of the Saliency Prediction Network used for our experiments***

### Additional details on the Saliency Predictor

The Saliency Predictor $S$ employs a ResNet-18 as encoder as to be similar to the paired classifier $C$, thus easing the attention-based modulation between the two branches. The saliency decoder is instead broadly inspired by UNISAL [155]. This architectural choice is motivated by the low number of parameters it requires, which leads to a short runtime if compared to other saliency models. In particular, the decoder consists of a stack of pointwise convolutions and deptwhise separable $3 \times 3$ convolutions, interleaved with bilinear upsampling blocks until the size of the original input image is recovered, while features from second and third residual blocks of the Encoder are used as skip connections, through two modules named *Skip-2* and *Skip-4*, to fuse features extracted at different abstraction levels. The architecture of the proposed model is illustrated in Fig. 6.3. Essentially, features from the bottleneck are upsampled with a factor $\alpha = 2$ and concatenated with the output of Skip-2 module. The obtained features maps are upsampled again with a factor $\beta = 2$ and concatenated with the output of Skip-4 module, while the number of feature maps is progressively scaled from the original value of 512 to 64. One last $1 \times 1$ convolution, followed by an upsampling layer and logistic activation, reduces the feature maps to 1 and the spacial sizes are restored to those of

the input image. More details are reported in Table 6.1.

| Saliency Model: Decoder | | | | | | |
|---|---|---|---|---|---|---|
| **Name** | **type** | **kernel/(stride)** | **Batch Norm** | **Activation** | **Input shape** | **Output shape** |
| *Post-cnn* | SepConv2D | $3 \times 3/(3,3)$ | Yes | ReLU | $512 \times 36 \times 48$ | $512 \times 36 \times 48$ |
| | Conv2D | $3 \times 3/(1,1)$ | Yes | — | $512 \times 36 \times 48$ | $256 \times 36 \times 48$ |
| *Upsampling-1* | Upsample $\alpha = 2$ | — | — | — | $256 \times 36 \times 48$ | $256 \times 72 \times 96$ |
| *Skip-2* | Conv2D | $1 \times 1/(1,1)$ | Yes | ReLU | $256 \times 72 \times 96$ | $512 \times 72 \times 96$ |
| | Conv2D | $1 \times 1/(1,1)$ | Yes | — | $512 \times 72 \times 96$ | $512 \times 72 \times 96$ |
| *Upsampling-2* | Conv2D | $1 \times 1/(1,1)$ | Yes | ReLU | $384 \times 72 \times 96$ | $768 \times 72 \times 96$ |
| | SepConv2D | $3 \times 3/(1,1)$ | Yes | ReLU | $768 \times 72 \times 96$ | $768 \times 72 \times 96$ |
| | Conv2D | $1 \times 1/(1,1)$ | Yes | — | $768 \times 72 \times 96$ | $128 \times 72 \times 96$ |
| | Upsample $\beta = 2$ | — | — | — | $768 \times 72 \times 96$ | $128 \times 144 \times 192$ |
| *Skip-4* | Conv2D | $1 \times 1/(1,1)$ | Yes | ReLU | $128 \times 144 \times 192$ | $256 \times 144 \times 192$ |
| | Conv2D | $1 \times 1/(1,1)$ | Yes | — | $256 \times 144 \times 192$ | $64 \times 144 \times 192$ |
| *Post-Upsampling-2* | Conv2D | $1 \times 1/(1,1)$ | Yes | ReLU | $192 \times 144 \times 192$ | $384 \times 144 \times 192$ |
| | SepConv2D | $3 \times 3/(1,1)$ | Yes | ReLU | $384 \times 144 \times 192$ | $384 \times 144 \times 192$ |
| | Conv2D | $1 \times 1/(1,1)$ | Yes | — | $384 \times 144 \times 192$ | $64 \times 144 \times 192$ |
| *Fusion* | Conv2D | $1 \times 1/(1,1)$ | — | Sigmoid | $64 \times 144 \times 192$ | $1 \times 144 \times 192$ |
| | Upsample $\gamma = 2$ | — | — | — | $1 \times 144 \times 192$ | $1 \times 288 \times 384$ |

***Table 6.1:*** *Detailed input-output sizes of the Decoder of our Saliency Prediction Network*

### Metrics and evaluation

As a primary metric of OCL model performance, we report the *final average accuracy* as $FAA = \frac{1}{T}\sum_{i=1}^{T} a_i^T$, where $a_i^T$ is the accuracy of the final model on the test set of task $\tau_i$. Accuracy $a_i^T$ can be computed in a *Class-Incremental Learning* (*Class-IL*) or in a *Task-Incremental Learning* (*Task-IL*) setting. In the latter, we assume that a task identifier is provided to the model at inference time, simplifying the problem by restricting the

set of class predictions for a given sample. While task-incremental learning is often depicted as a trivial scenario in recent literature [68, 8, 34], we emphasize its usefulness, as it isolates the effect of within-task forgetting from the model's bias towards the currently learned classes [65, 64, 86]. For this reason, we report both Class-IL and Task-IL performance in the results. Results are reported in terms of mean and standard deviation over five different runs.

### 6.4.3 Results

| Model | Seq-Mini-ImageNet | | | Seq-FG-ImageNet | | |
|---|---|---|---|---|---|---|
| *Class-incremental learning* | | | | | | |
| Joint | 14.79±1.17 | | | 9.06±1.07 | | |
| ↪**SAM** | **16.10**±0.30 | | | **9.73**±0.73 | | |
| Fine-tune | 3.43±0.35 | | | 2.43±0.81 | | |
| ↪**SAM** | **4.20**±0.27 | | | **3.68**±0.44 | | |
| **Buffer size** | **1000** | **2000** | **5000** | **1000** | **2000** | **5000** |
| DER++ | 14.95±3.11 | 12.82±4.97 | 14.58±2.55 | 8.08±1.54 | 8.27±1.72 | 9.20±0.86 |
| ↪**SAM** | **19.13**±1.62 | **22.92**±2.25 | **25.35**±2.56 | **11.71**±2.36 | **12.97**±1.62 | **13.73**±1.95 |
| ER-ACE | 20.86±3.69 | 24.93±3.20 | 26.31±5.22 | 14.28±0.96 | 16.45±1.24 | 18.21±3.45 |
| ↪**SAM** | **27.48**±2.83 | **33.09**±1.28 | **35.58**±1.79 | **20.03**±3.13 | **23.80**±2.11 | **28.68**±0.50 |
| CoPE | 21.58±1.60 | 23.58±4.39 | 24.77±3.56 | 16.45±1.38 | 16.81±0.83 | 17.77±2.02 |
| ↪**SAM** | **26.66**±2.22 | **33.35**±4.67 | **45.04**±2.44 | **18.17**±2.79 | **27.14**±1.62 | **34.34**±3.51 |
| *Task-incremental learning* | | | | | | |
| Joint | 63.12±1.19 | | | 56.33±2.51 | | |
| ↪**SAM** | **64.18**±0.60 | | | **56.72**±1.09 | | |
| Fine-tune | 34.08±2.28 | | | 28.81±1.66 | | |
| ↪**SAM** | **57.07**±3.44 | | | **51.24**±2.36 | | |
| DER++ | 73.07±3.07 | 75.11±5.61 | 77.71±3.04 | 68.65±2.14 | 70.24±3.97 | 74.74±1.14 |
| ↪**SAM** | **79.75**±1.56 | **82.97**±0.25 | **84.10**±0.81 | **72.83**±3.90 | **75.40**±2.29 | **78.26**±1.10 |
| ER-ACE | 71.00±3.21 | 75.60±3.47 | 77.17±4.08 | 66.27±0.92 | 69.09±3.15 | 70.88±5.72 |
| ↪**SAM** | **77.51**±2.72 | **82.22**±0.96 | **83.56**±1.55 | **73.08**±2.14 | **75.60**±2.28 | **79.46**±0.56 |
| CoPE | 68.00±0.73 | 71.76±2.95 | 74.31±2.25 | 63.77±2.32 | 67.29±3.33 | 69.14±2.93 |
| ↪**SAM** | **72.69**±0.80 | **77.57**±1.57 | **84.64**±1.20 | **64.79**±1.60 | **73.39**±1.11 | **78.66**±1.59 |

***Table 6.2:*** *Final Average Accuracy FAA [↑] in Class-IL and Task-IL for rehearsal-based methods with and without SAM.*

### OCL performance

We first evaluate the contribution that attention-driven modulation provides to state-of-the-art OCL baselines. For each method, we compute class-incremental and task-incremental accuracy and compare to those obtained when integrating SAM, as described in Sect. 6.3.

Results for rehearsal methods are reported in Table 6.2, showing a pattern of enhanced performance when integrating SAM, for all tested buffer sizes. Table 6.3 shows results for non-rehearsal methods. In this case, SAM improvements are more evident in task-incremental; a marginal gain in class-incremental accuracy is also noticeable, though the low performance of baselines limits the room for improvements.

| **Model** | **Seq-Mini-ImageNet** | | **Seq-FG-ImageNet** | |
|---|---|---|---|---|
| | CLASS-IL | TASK-IL | CLASS-IL | TASK-IL |
| Joint | 14.79±1.17 | 63.12±1.19 | 9.06±1.07 | 56.33±2.51 |
| ↪**SAM** | **16.26**±0.30 | **64.34**±0.59 | **9.51**±0.93 | **56.72**±1.09 |
| Fine-tune | 3.43±0.35 | 34.08±2.28 | 2.43±0.81 | 28.81±1.66 |
| ↪**SAM** | **4.20**±0.27 | **57.07**±3.44 | **3.68**±0.44 | **51.24**±2.36 |
| LwF | 3.18±0.41 | 30.61±1.80 | 3.25±0.45 | 27.55±1.64 |
| ↪**SAM** | **4.22**±0.31 | **48.61**±2.14 | **3.57**±0.23 | **36.57**±2.09 |
| oEwC | 2.68±0.24 | 24.10±1.55 | 2.38±0.23 | 24.98±1.15 |
| ↪**SAM** | **3.08**±0.31 | **35.33**±3.18 | **2.55**±0.55 | **26.02**±1.64 |

***Table 6.3:*** *Final Average Accuracy FAA [↑] in Class-IL and Task-IL for non-rehearsal methods with and without SAM.*

Since our strategy foresees two paired networks for classification and

saliency prediction, we also compare with similar multi-branch CL baselines:

- **DualNet** [43], mentioned in Sect. 6.1, employs a dual-backbone architecture to decouple incremental classification (by a *fast learner*) from self-supervised representation learning [115] (by a *slow learner*). We adapt SAM to DualNet by replacing the slow learner and its training objective with our saliency prediction backbone, forcing the fast learner to use saliency features for classification.
- **TwF** [156] employs a frozen pre-trained classification backbone to stabilize the learning of class-incremental features, by means of an attention mechanism. To enable SAM, the pre-trained classification backbone and the feature distillation strategy are replaced with the saliency encoder, and the features of the two backbones are combined through multiplication, as described in Sect. 6.3.

Table 6.4 shows results for different buffer sizes (we could not run TwF with buffer size of 5000, due to excessive computing requirements). Integrating SAM outperforms baseline versions of both methods, suggesting that controlling learning through visual attention leads to better representation for classification than, for instance, contrastive learning. This is inline with cognitive neuroscience findings [146, 187], for which object identity-preservation, that also involves contrastive learning, happens mostly at later layers (e.g., IT neurons), while selective attention acts during the whole categorization process.

### Effect of classification pre-training

Additionally, in order to demonstrate generalization capabilities of our attention-modulated strategy, and to ground our approach to the CL meth-

| Buffer | Model | Seq-Mini-ImageNet | | Seq-FG-ImageNet | |
|---|---|---|---|---|---|
| | | CLASS-IL | TASK-IL | CLASS-IL | TASK-IL |
| 1000 | TwF | 23.78±1.67 | 73.57±1.27 | 15.32±2.59 | 64.32±5.18 |
| | ↪**SAM** | **28.36**±3.72 | **79.28**±2.24 | **20.04**±1.63 | **71.35**±1.70 |
| | DualNet | 20.57±0.91 | 72.65±0.56 | 15.62±1.54 | 67.60±1.56 |
| | ↪**SAM** | **28.58**±1.40 | **81.79**±0.59 | **19.48**±0.59 | **75.76**±0.51 |
| 2000 | TwF | 29.05±2.02 | 78.38±1.66 | 18.72±1.75 | 72.15±2.82 |
| | ↪**SAM** | **35.55**±0.61 | **82.98**±0.85 | **22.54**±2.20 | **73.34**±2.94 |
| | DualNet | 27.41±1.79 | 76.49±0.65 | 21.04±1.08 | 71.54±0.72 |
| | ↪**SAM** | **33.76**±1.21 | **83.79**±0.27 | **22.53**±1.56 | **78.35**±0.36 |
| 5000 | DualNet | 32.08±1.55 | 80.26±0.97 | 22.07±2.08 | 74.53±1.27 |
| | ↪**SAM** | **36.44**±0.77 | **85.72**±0.40 | **24.83**±2.01 | **80.18**±0.52 |

***Table 6.4:*** *Comparison of our saliency-attention mechanism to computational attention mechanisms (TwF [156]) and contrastive learning (DualNet [43]) for stabilizing learned classification features in CL tasks.*

ods that exploit pre-training, we also compute performance when the classifier backbone and saliency encoder are pre-trained on a classification pretext task (despite using classification-pretrained fetures appears to be in contrast to what it happens in the human brain). Differently from what describe in 6.4.2, here we use the same disjoint subset of ImageNet classes to train the backbone of the classifier, then we initialize the saliency Encoder to the same weights.

Also in this setting, methods combined to SAM achieve better results, as show in Table 6.5. However, the performance gain is lower than the

| Model | Seq-Mini-ImageNet | | | Seq-FG-ImageNet | | |
|---|---|---|---|---|---|---|
| Buffer | 1000 | 2000 | 5000 | 1000 | 2000 | 5000 |
| | CLASS-IL | | | CLASS-IL | | |
| DER++ | 30.35±0.74 | 30.96±0.59 | 32.55±1.47 | 15.76±0.58 | 16.61±0.26 | 16.83±0.44 |
| ↪**SAM** | **31.20**±2.39 | **33.91**±2.31 | **37.91**±1.07 | **17.06**±1.51 | **20.43**±2.11 | **22.53**±0.82 |
| ER-ACE | 42.33±0.57 | 45.84±0.50 | 48.77±1.28 | 30.91±1.02 | 34.09±0.57 | 37.49±0.47 |
| ↪**SAM** | **46.56**±1.10 | **50.52**±0.69 | **53.23**±0.35 | **32.46**±1.09 | **36.08**±1.60 | **40.73**±0.84 |
| | TASK-IL | | | TASK-IL | | |
| DER++ | **89.98**±0.75 | **91.14**±0.20 | **91.37**±0.10 | **83.87**±0.81 | **85.61**±0.29 | **86.19**±0.21 |
| ↪**SAM** | 89.34±0.54 | 90.47±0.32 | 91.36±0.30 | 82.34±0.54 | 84.04±0.40 | 84.83±0.32 |
| ER-ACE | 88.28±0.50 | 90.14±0.05 | 91.23±0.13 | 82.83±0.40 | **85.39**±0.38 | **87.29**±0.08 |
| ↪**SAM** | **89.99**±0.46 | **90.83**±0.20 | **91.84**±0.08 | **82.94**±1.15 | 84.25±0.95 | 86.51±0.25 |

***Table 6.5:*** *Final Average Accuracy FAF [↑] in Class-IL and Task-IL when the classifier backbone and saliency encoder are pre-trained on a classification task with classes different from those available in the CL settings.*

one obtained with saliency pre-training. This is possibly due the fact that classification pre-trained features are better than saliency one (as also evidenced by the general higher performance obtained with classification pre-training) and have reched their maximum capacity. Thse results confirm again the contribution of the *forgetting-free* behaviour of the saliency prediction task to classification task.

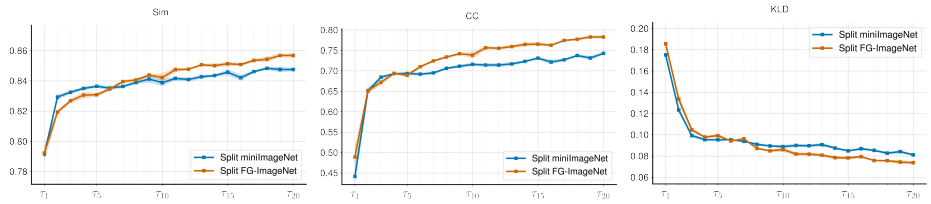



***Figure 6.4: Saliency prediction accuracy**, measured in terms of Similarity (SIM), Pearson's Correlation Coefficient (CC) and Kullback-Leibler divergence (KLD) metrics, in continual learning settings on the Seq-Mini-Imagenet and Seq-FG-ImageNet benchmarks.*

### Saliency Prediction vs Attribution Maps

We also compute the saliency metrics obtained by our saliency predictor $S$, in the considered class-incremental setting. In particular, we use three widely used metrics for image saliency prediction [145]: Persons's Correlation Coefficient (CC), Similarity (Sim) and Kullback-Leibler divergence (KLD). As shown in Fig. 6.4, all metrics do not degrade as new tasks are processed, but rather they exhibit a trend of enhancement with the number of CL tasks.

This behaviour is further corroborated by the qualitatively results shown in Fig. 6.5. The predicted saliency maps show no significant forgetting when training on a sequence of twenty tasks (from $\tau_0$ to $\tau_{20}$).

Conversely, the attribution maps computed through Grad-CAM [163] significantly deteriorates, showing a high level of forgetting. These results thus demonstrate that pairing a saliency prediction model with a classifier yields better results than storing attribution maps as done in [162].


***Figure 6.5: Qualitative comparison of attribution maps computed through GradCAM (first row) and the saliency maps produced by the saliency predictor*** $S$ ***(second row) during a continual training on a sequence of 20 tasks.*** *GradCAM attributions maps show significant forgetting, while saliency maps tend steadily to improve while training.*

### Cost analysis

We finally perform cost analysis to assess the efficiency of our SAM approach compared to existing methods that employ two branches, i.e., TwF [156] and DualNet [43]. It is important to note that in a continual learning settings, efficiency at training time might be more relevant than the one at inference times as the main assumption is of a deep model that keeps training from an infinite strem of data. The comparison in carried out on an NVIDIA A100 and using the ResNet18 backbone for all models. The results in Table 6.6 reveals that SAM is much more efficient than DualNet and TwF at training time, while it shows higher costs at inference time (but also an accuracy gain of $\sim$10 points).

| Metric | DualNet [43] | TwF [156] | SAM |
|---|---|---|---|
| Train parameters | 16 M | 58 M | 23 M |
| Train time | $\sim$ 6.5 h | $\sim$ 3.0 h | $\sim$ 1.0 h |
| Inference parameters | 16 M | 11 M | 22 M |
| Inference time | 3.45 ms | 3.15 ms | 7.50 ms |

***Table 6.6:*** *Efficiency analysis. Comparison of training and inference times and parameters between SAM, DualNet and TwF.*

### 6.4.4 Ablation Studies

The proposed strategy is grounded on cognitive neuroscience literature, according to which selective attention modulates neuronal responses of all layers involved in the categorization process, in a multiplicative fashion. Our next experiments are meant to assess whether this hypothesis (i.e., feature modulation through multiplication for all classification layers) is optimal also for artificial neural networks, or if other integration modalities of saliency information may be equally effective. We thus compare our SAM strategy with the following baselines, all exploiting saliency information in different ways:

- **Saliency-based input modulation (SIM)**: the input image is multiplied by the corresponding estimated saliency map (thus highlighting salient regions only).
- **Saliency as additional input (SAI)**: we modify the classification network to receive as input a 4D data tensor, with the saliency map concatenated to RGB channels.
- **Learning saliency-based modulation (LSM)**: rather than multiplying

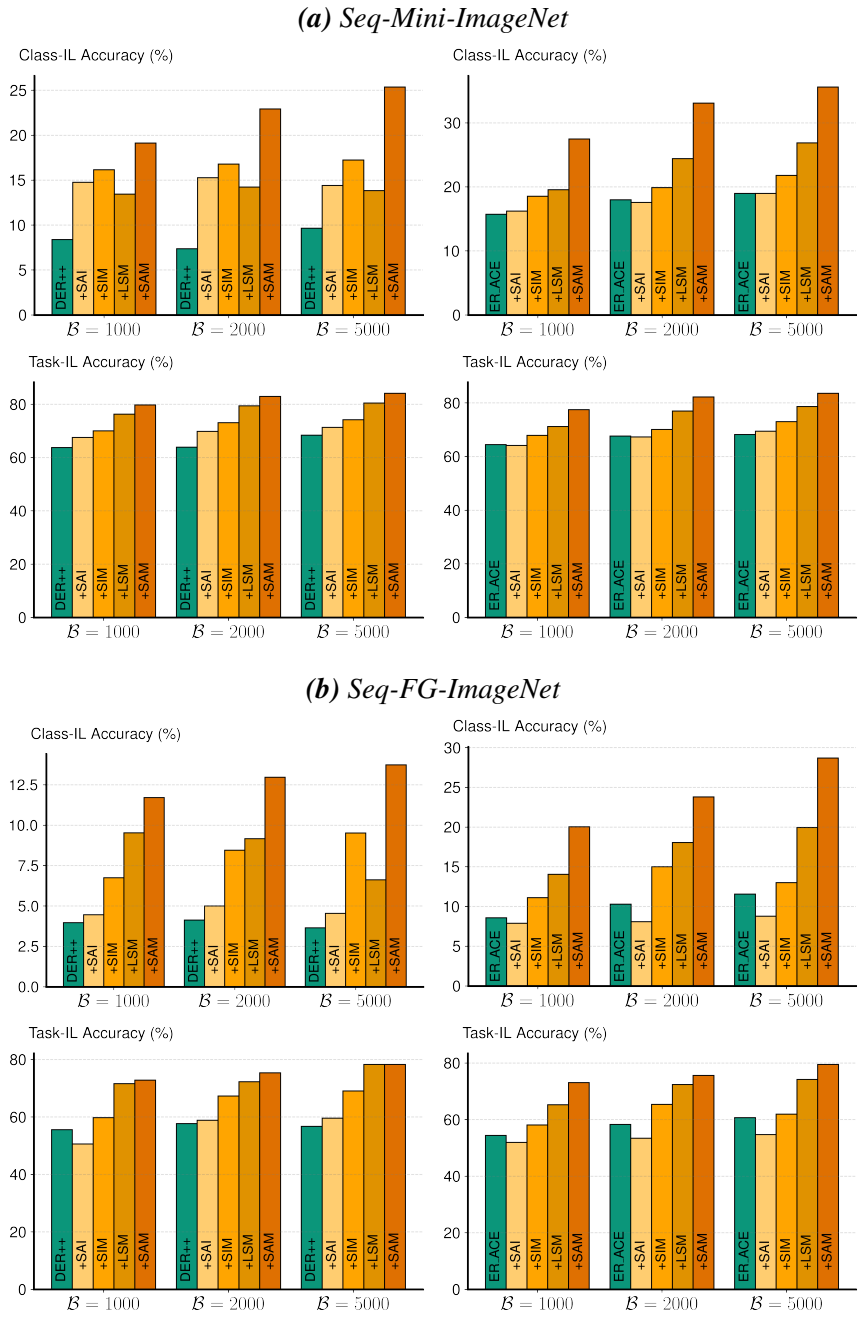



***Figure 6.6: Comparison of SAM to alternative saliency integration strategies.*** ***SIM*** *modulates input images by saliency maps.* ***SAI*** *provides saliency maps as an additional input channel to the classification network.* ***LSM*** *merges classification and saliency features through a learnable convolutional layer.*

classification features $\mathbf{z}_{i-1}^{(c)}$ and saliency features $\mathbf{z}_{i-1}^{(s)}$ (see Eq. 6.4), we feed them to convolutional layer with $1 \times 1$ kernel to produce $\mathbf{z}_{i}^{(c)}$, and let the model learn the corresponding parameters.

Fig. 6.6 reports the results of this analysis, using DER++ and ER-ACE as baseline methods, and clearly indicates the superiority the SAM strategy to other saliency integration variants. However, it is interesting to note that saliency helps classification performance in all cases, demonstrating its usefulness for continual learning tasks. We argue that this is due to the intrinsic nature of saliency prediction, which we found to be i.i.d. with respect to the data stream.

We then investigate whether the impact of attention-driven modulation is uniform across the backbone layers. To this aim, we define a positional binary coding scheme, controlling the application of the SAM strategy at the predefined points of the network (see Sect. 6.4.2): if position $i$ of the coding scheme is 1, then the $i$-th feature modulation point is enabled, i.e., features from the $i$-th block of the classification network are multiplied by the features of the $i$-th block of the saliency network. Results are reported in Table 6.7 for both DER++ and ER-ACE, and indicate that the best strategy is to modulate the features of all classification layers through the corresponding saliency ones, similarly to what neurophysiological evidence reports [151, 152].

### 6.4.5 Model Robustness

We finally assess the robustness of the SAM strategy in dealing with *spurious features* and *adversarial attacks*. Spurious features are information that correlates well with labels in training data but not in test data (e.g., in a classification task between birds and dogs, training with yellow birds and

| SAM Scheme | Seq-Mini-ImageNet | | Seq-FG-ImageNet | |
|---|---|---|---|---|
| | Class-IL | Task-IL | Class-IL | Task-IL |
| | | *DER++* | | |
| **1 1 1 0 0** | 12.97±2.62 | 74.55±3.62 | 6.54±0.67 | 67.34±1.38 |
| **1 1 1 1 0** | 17.46±1.02 | 80.15±0.34 | 8.77±1.45 | 71.51±2.92 |
| **1 1 1 1 1** | **22.92**±2.25 | **82.97**±0.25 | **12.97**±1.62 | **75.40**±2.29 |
| | | *ER-ACE* | | |
| **1 1 1 0 0** | 23.72±0.77 | 74.15±1.38 | 18.08±0.96 | 70.44±2.08 |
| **1 1 1 1 0** | 26.44±2.33 | 77.14±2.73 | 16.55±2.55 | 67.32±5.07 |
| **1 1 1 1 1** | **33.09**±1.28 | **82.22**±0.96 | **23.80**±2.11 | **75.60**±2.28 |

***Table 6.7:*** *Performance comparison in term of FAA [↑] when applying SAM to DER++ and ER-ACE at different layers of the ResNet-18 backbone, with buffer size 2000.*

black dogs only), leading to low generalization [188]. This effect is exacerbated in continual learning settings, where the covariate shift between train data and test data increases as new tasks come in. Thus, we measure to what extent our SAM strategy can mitigate the tendency of learning methods to exploit spurious features to solve classification tasks.

We crafted an ad-hoc benchmark consisting of ten classes from ImageNet. For each class, we added a class signature for training images, leaving the test images unaltered. In detail, we modified each training image by increasing the brightness of all pixels by a class-dependent offset, computed as $5(c+1)$ (in a 0-255 brightness range), where $c$ is a numeric class label. We then define five continual learning tasks with two classes

| **Method** | Class-IL | Task-IL |
|---|---|---|
| ER-ACE | $50.07_{\pm 3.88}$ | $86.77_{\pm 1.63}$ |
| ER-ACE$^{\mathcal{SF}}$ | $28.46_{\pm 3.46}$ | $74.40_{\pm 4.37}$ |
| $\hookrightarrow$**SAM** | $\mathbf{44.08}_{\pm 3.67}$ | $\mathbf{83.04}_{\pm 3.06}$ |

***Table 6.8:*** *Effect of the SAM strategy in the presence of spurious features. The $\mathcal{SF}$ apex indicates that the method is trained on the biased dataset containing spurious features, while the one without apex when ER-ACE it trained on the original, spurious-free, dataset.*

each. We then compare ER-ACE to the corresponding SAM-enabled variant and ground its performance with the one obtained when it is trained with original images (i.e., without enforcing spurious features in the data). Results in Table 6.8 show that SAM effectively limits the possibility for the classifier to use spurious features, resulting in a more robust and generalizing model. The drop of performance (about 22 percent points) observed between training with the original data and training with data biased by spurious features is almost completely recovered when SAM is used.

Finally, we evaluate the robustness of SAM against adversarial perturbations of the input space. To this aim, we apply the Projected Gradient Descent (PGD) attack [189] with different $\varepsilon$ values (determining the strength of the attack) and compare the average performance drop experienced by ER-ACE, in its original version and when combined with SAM. We conduct the evaluation on both Seq-Mini-ImageNet and Seq-FG-ImageNet, repeating each experiment three times. As shown in Figure 6.7, SAM considerably improves model stability, counteracting perturbations by regularizing classification features with saliency ones.

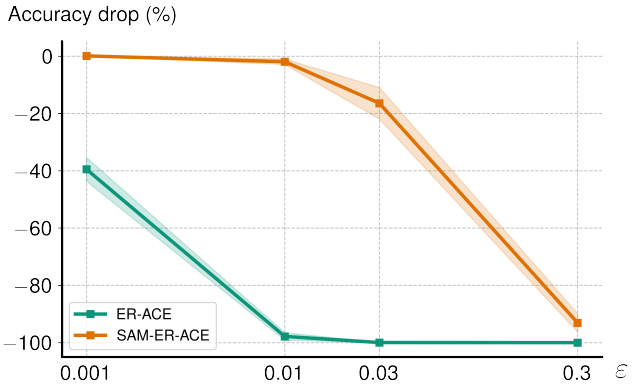



***Figure 6.7: Robustness to adversarial attacks**. ER-ACE baseline drops even with small attacks, while SAM significantly enhances robustness.*

## 6.5 Discussion

We presented SAM, a biologically-inspired selective attention-driven modulation strategy for online continual learning, which regularizes classification features using visual saliency, effectively reducing forgetting. The proposed approach, grounded on neurophysiological evidence, significantly improves performance of state-of-the-art OCL methods, and has been shown to be superior to other multi-branch solutions, either biologically-inspired (e.g., DualNet) or based on feature attention mechanisms (e.g., TwF).

Our results confirm that adapting neurophysiological processes into current machine learning techniques is a promising direction to bridge the gap between humans and machines. Future research directions will address both limitations and extensions of the proposed approach. Indeed, while SAM is model-agnostic, its formulation requires that the saliency encoder

and the classifier are architecturally identical. The application to heterogeneous networks will be explored by defining or learning a mapping between activations at different network stages. Moreover, our finding that saliency prediction is i.i.d. with respect to classification distribution shifts will lead to investigate whether other low-level visual tasks enjoy this property.

## 6.6 Publications

The approach described in this chapter is currently under review at IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) journal.

Bellitto, G., Proietto Salanitri, F., Pennisi, M., Boschini, M., Bonicelli, L., Porrello, A., Calderara. S., Palazzo, S., Spampinato C., 2023. Selective Attention-based Modulation for Continual Learning. Submitted to IEEE Transaction on Pattern Analysis and Machine Learning.

CHAPTER

# SEVEN

# WAKE-SLEEP CONSOLIDATED LEARNING

In this final step of our journey, we leverage the Complementary Learning System theory, already discussed in the previous Chapter 6, and complement it with theories on off-line brain states to propose another neuro-inspired method able to reduce forgetting in CL settings.

Sleep, which appears to be a passive state, is actually a dynamic process that plays a crucial role in various cognitive functions in humans. Research over the past few decades has highlighted the role of sleep in memory consolidation [190]. During sleep, neural representation of new experiences are strengthened and integrated into existing knowledge bases. Sleep ensures that learning is not just a fleeting event that exclusively occurs during wake stage, when an agent experiments with new visual stimuli, but that learning is ingrained and retained over time during sleep, which facilitates the consolidation of newly encoded information through unique neuromodulatory activities.

This chapter draws an intriguing parallel between the benefits of sleep

in humans and the desire to maintain and consolidate prior knowledge in continual learning. Taking inspiration from the underlying mechanisms of wake-sleep memory acquisition and consolidation in humans, we aim to glean insights that can inform and inspire novel strategy for knowledge preservation in AI systems.

## 7.1 Motivation

Humans and machines learn in different ways: the inherent structure and optimization approaches of artificial neural networks differs significantly from how humans build neural connectivity over a lifetime. The Complementary Learning Systems (CLS) theory [191, 192] suggests that effective human learning occurs through the interplay of two learning processes originating from the hippocampus and neocortex brain regions. These regions interact to learn representations from experience (neocortex) while consolidating and sustaining long-term memory (hippocampus). This theory has inspired continual learning methods [43, 193] which translate CLS concepts into computational frameworks. DualNet [43] employs two learning networks: a slow learner that emulates the memory consolidation process in the hippocampus and a fast learner that adapts current representations to new observations. DualPrompt [193] addresses the challenge of adapting transformer models to new tasks while minimizing the loss of previous knowledge, using learnable prompts that are responsible for adapting to new data quickly, while preventing catastrophic forgetting. The specialization of prompt sets to their respective tasks is similar to how the hippocampus and neocortex specialize in complementary learning processes. DualNet and DualPrompt suggest that grounding ar-

tificial neural networks to cognitive neuroscience may result in improved performance, as they both achieve state-of-the-art performance on multiple benchmarks. Though promising, these approaches are rather rigid as the structures of the two learning parts (network architecture in DualNet; prompt format and positioning in DualPrompt) are defined *a priori*, while neural networks in primates perform fast adaptation by flexibly reconfiguring synapses while learning from new experience. Moreover, prior work does not consider the role of offline brain states such as sleep. Current theories suggest that sleep and dreaming play a crucial role in consolidating memories and facilitating learning, by increasing generalization of knowledge [194, 195, 196]. During sleep, neurons are spontaneously active without external input and generate complex patterns of synchronized activity across brain regions [197, 198]. This strong neural activity is believed to be due to the brain replaying and consolidating memories, while reorganizing synaptic connections.

In this work we propose Wake-Sleep Consolidated Learning (**WSCL**), extending the CLS theory by including wake-sleep states, in order to improve artificial neural networks' continual learning capabilities. This integration is achieved by introducing a sleep phase at training time that mimics the offline brain states during which synaptic connection, memory consolidation and dreaming occur. In WSCL, a deep neural network (DNN) replicates the functions of the neocortex, while a two-layered buffer for short-term and long-term memory mimics the role of the hippocampus. Training is organized in two main phases: 1) a *wake phase*, where fast adaption of the DNN to new sensory experience is carried out and episodic memories are stored in the short-term memory; 2) a *sleep phase*, consisting of two alternating stages: a) Non-Rapid Eye Movement (NREM), where the network replays episodic memories collected during the wake step,

consolidates past experiences in the long-term memory, and optimizes its neural connections to support synaptic plasticity; b) Rapid Eye Movement (REM), where dreaming simulates new experience, preparing the brain for future events. The hypothesis is that dreaming serves as an "anticipatory" mechanism, helping the brain to identify relationships between different types of information and making it easier to learn and remember new information.

Our computational formulation of the week-sleep process is tested on several benchmarks, including CIFAR-10, Tiny-ImageNet and FG-ImageNet. In all cases, our method outperforms the baselines and prior work, yielding a significant gain in classification tasks. Remarkably, WSCL approach is the first continual learning method yielding positive forward transfer, demonstrating its ability to prepare synapses to future knowledge. We also show that all three steps are necessary: the wake stage is essential to ensure efficiency and to favor network plasticity by the NREM stage, while the REM stage helps to increase feature transferability and reduce the forgetting of acquired knowledge.

In summary, we make the following contributions:

- We present Wake-Sleep Consolidated Learning (WSCL), a novel continual learning framework for enhancing neural networks capability by incorporating wake-sleep states, inspired by the brain's offline activities.

- We propose our WSCL framework which consist of two primary phases: a wake phase focused on rapid adaptation to new experiences and memory storage, and a sleep phase with Non-Rapid Eye Movement (NREM) stage for memory consolidation, interspersed with a Rapid Eye Movement (REM) stage that simulate optimiza-

tion and anticipation.

- We conduct extensive experiments to demonstrate the effectiveness of our WSCL, and we discover that methods trained with WSCL achieve superior results across various challenging benchmarks. Noteworthy, WSLC has a positive impact on forward transfer, proving that all the three phases are essential in finding the proper trade-off between retention of prior knowledge and adaptation to future tasks.

## 7.2 Related Work

Continual Learning (CL) [18, 46] is a branch of machine learning whose objective is to bridge the gap in incremental learning between humans and neural networks. McCloskey and Cohen [5] highlight that the latter undergo *catastrophic forgetting* of previously acquired knowledge in the presence of input distribution shifts. To mitigate this problem, several solutions have been proposed, introducing either adequate regularization terms [23, 25], specific architectural organization [24, 27] or the rehearsal of a small number of previously encountered data points [32, 31, 38].

While current solutions help reducing forgetting, real-world application proves difficult, as typical CL evaluations are carried out on unrealistic benchmarks [157, 158]. Most approaches tackling this challenging scenario combine a replay strategy [6, 32, 47] to regularization on logits sampled throughout the optimization trajectory [38]. Some works focus on memory management: GSS [34] introduces a specific optimization of the basic rehearsal formula meant to store maximally informative samples; HAL [126] individuates synthetic replay data points that are maxi-

mally affected by forgetting. Other works propose tailored classification schemes: CoPE [121] uses class prototypes to ensure a gradual evolution of the shared latent space; ER-ACE [40] makes the cross-entropy loss asymmetric to minimize imbalance between current and past tasks. Recent works introduce a surrogate optimization objective: CR [159] employs a supervised contrastive learning objective and OCM [160] leverages mutual information: both aim at learning features that are less subject to forgetting.

Our approach differs from these classes of methods, in that we take inspiration from cognitive neuroscience theory of learning (Complementary Learning Systems and wake-sleep) and exploits brain off-line states such as sleeping and dreaming. We demonstrate that alternating standard training with a revisited strategy that combines on-line and off-line stages makes the model more resilient to task shifts. Recently, a few neuroscience-informed CL methods have been proposed. Elastic Weight Consolidation (EWC) [23] and Synaptic Intelligence [25] employ regularization to preserve important weights learned during previous tasks while allowing the network to adapt to new tasks, emulating fast adaption happening in the neocortex. FearNet [144] adopts an auxiliary network (in line with CLS theory) to detect catastrophic forgetting and trigger knowledge-preserving regularization. Co2L [41] learns stable representations through contrastive learning and self-supervised distillation.

Two approaches similarly inspired by CLS theory are DualNet [43] and DualPrompt [193]. DualNet employs two networks that loosely emulate slow and fast learning in humans. DualPrompt [193] also takes a cognitive approach, using learnable prompts to be paired to a pretrained transformer backbone. While both approaches yield good results, they ignore off-line states, that appear fundamental in human learning. Though

not applied to continual learning yet, the wake-sleep algorithm has been shown to have the potential for learning improved and robust semantic representations [199, 200]. Another related approach is Sleep Replay Consolidation [201] that employs sleep-based training using local unsupervised Hebbian plasticity rules for mitigating catastrophic forgetting of ANN.

WSCL further unfolds the sleep phase by detailing the NREM and REM stages, integrating the dreaming process into the learning loop. This integration, which appears to contribute significantly to human learning, has a positive impact on the training of neural networks (as shown in the results). The computational formulation of the wake-NREM-REM of WSCL is inspired by [202], where the role of adversarial dreaming for learning visual representations is preliminary investigated. However, simple strengthening of existing connections through unsupervised learning as proposed in [201, 202] does not seem sufficient to build robust representations during sleep [196]: our work thus explores more sophisticated restructuring of neural connections in the neocortex guided by the hippocampus.

## 7.3 Method

An overview of the WSCL approach is presented in Fig. 7.1, showing how the training stage on a new task is divided into two phases: a *wake phase* and a *sleep phase*.

During the wake phase, the model is exposed to the new task, with the objective of performing fast adaptation of existing knowledge to the task characteristics. In this stage, the model quickly updates its parameters in order to find a balance between previously-acquired knowledge and new information, storing the latter in a short-term memory for later reuse

during the sleep stage. In implementation terms, this balance is achieved by dynamically and adaptively freezing layer representations, identifying plasticity requirements for learning the new task while enforcing stability. Thus, during the wake stage, WSCL focuses primarily on learning general and transferable representation by combining both current and past experience. In the sleep phase, the model consolidates newly acquired knowl-

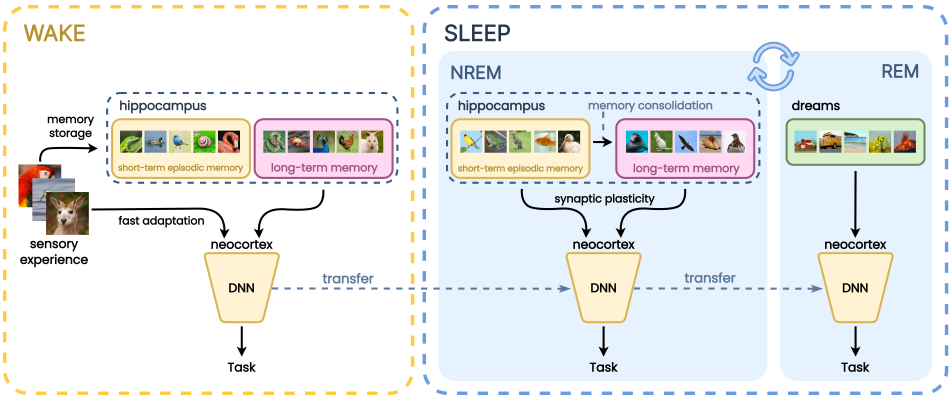



***Figure 7.1: Wake-Sleep Consolidated Learning****: in the wake stage, the model (which emulates the neocortex) fast adapts to the new sensory experience, storing episodic memories (as in the hippocampus) in the short-term memory to be replayed during sleep. The sleep phase foresees two alternating processes: 1) the NREM stage, where the DNN model consolidates its synapses based on the replayed (recent and past) samples and the long-term memory is updated; 2) the REM stage, where the DNN is trained with dreamed samples to prepare the model for future sensory inputs.*

edge by revisiting the hippocampus short-term memory containing the task data, merging it into existing knowledge by updating synaptic connections, moving it into a long-term memory for future reference, and exploring the

representational space through task-agnostic "dreaming". These stages are mapped into our training procedure by means of supervised training on task data, buffering task information in a (small) long-term memory, and employing an auxiliary dataset (uncorrelated to task data) as a surrogate for the generative process associated to dreaming.

### 7.3.1 Problem formulation

Following the established literature, we pose continual learning as a supervised classification problem on a non-i.i.d. stream of data, with the assumption that *task boundaries*, marking changes in the data distributions, are known at training time. More formally, let $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$ be a sequence of data streams, where each pair $(\mathbf{x}, y) \sim \mathcal{D}_i$ denotes a data point $\mathbf{x} \in \mathcal{X}$ with the corresponding class label $y \in \mathcal{Y}$; the sample distributions (in terms of both the data point distribution and the class label distribution) of different $\mathcal{D}_i$ and $\mathcal{D}_j$ may vary — for instance, class labels from $\mathcal{D}_i$ might be different from those from $\mathcal{D}_j$. Given a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$, parameterized by $\boldsymbol{\theta}$, the objective of continual learning is to train $f$ on $\mathcal{D}$, organized as a sequence of $T$ tasks $\{\tau_1, \ldots, \tau_T\}$, under the constraint that, at a generic task $\tau_i$, the model receives inputs sampled from the corresponding data distribution only, i.e., $(\mathbf{x}, y) \sim D_i$. The classification model may also keep a limited *memory buffer* $\mathbf{M}$ (assumed to be our long-term memory in the hippocampus) of past samples, to reduce forgetting of features from previous tasks. The model update step between tasks can be summarized as:

$$\langle f, \boldsymbol{\theta}_{i-1}, \mathbf{M}_{i-1} \rangle \xrightarrow{\mathcal{D}_i} \langle f, \boldsymbol{\theta}_i, \mathbf{M}_i \rangle \tag{7.1}$$

where $\boldsymbol{\theta}_i$ and $\mathbf{M}_i$ represent the set of model parameters and the memory buffer at the end of task $\tau_i$.

The training objective is to optimize a classification loss over the sequence of tasks (without losing accuracy on past tasks) by the model instance at the end of training:

$$\arg\min_{\boldsymbol{\theta}_T} \sum_{i=1}^{T} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i} \Big[ \mathcal{L}\Big( f\left(\mathbf{x}; \boldsymbol{\theta}_T\right), y \Big) \Big] \tag{7.2}$$

where $\mathcal{L}$ is a generic classification loss (e.g., cross-entropy), which a continual learning model attempts to optimize while accounting for model *plasticity* (the capability to learn current task data) and *stability* (the capability to retain knowledge of previous tasks) [5].

### Wake phase

According to the established cognitive foundation, we define the waking stage in the proposed learning paradigm as the combination of two simultaneous processes, *short-term memorization* and *fast model adaptation*.
**Short-term memorization** has the objective of storing part of the current task experience, for later reuse — in particular, for processing and consolidation during the sleep stage. In a continual learning setting, we model short-term memorization into $\mathbf{M}_s$ as a sampling of task data $\mathcal{D}_i$:

$$\mathbf{M}_s = \{(\mathbf{x}_j, y_j) \sim \mathcal{D}_i\}_{j=1}^{N_s}, \tag{7.3}$$

where $N_s$ is the amount of samples collected from the $\mathcal{D}_i$ distribution[1]. Note that $\mathbf{M}_s$ is reset during each wake phase and is distinguished from the *long-term memory* $\mathbf{M}_i$, which includes a smaller permanent number

[1]For brevity, we drop task index $i$ from short-term memory $\mathbf{M}_s$, as it is re-created at each task.

of samples $N_l$ from past tasks (in practice, the buffer of rehearsal-based methods).

### Fast model adaptation

In accordance to CLS theory [191, 192], we propose a method for fast model adaptation that employs parameter freezing during the wake stage to maximize stability and plasticity. Specifically, we propose to train the model for a limited number of iterations under varying parameter freezing settings, providing an opportunity to the model to rapidly learn new information in the wake stage while retaining the previous knowledge; in-depth consolidation of task information will be carried out separately in the sleep stage. Unlike approaches such as DualNet, where the structure of the slow and fast networks are predefined, in WSCL the part of the network that reuses past knowledge and the part accounting for plasticity are identified on-line during the wake phase.

Formally, we want to model the joint probability between task data $\mathcal{D}_i$, previous experience $\mathbf{M}_{i-1}$, model parameters $\boldsymbol{\theta}_i$ and a binary freezing mask $\mathbf{m}_i$, with the same dimensions as $\boldsymbol{\theta}_i$ and such that $m_{i,j} = 1$ indicates that parameter $\theta_{i,j}$ should be frozen:

$$P(\mathbf{x}, y, \boldsymbol{\theta}_i, \mathbf{m}_i) = P\left(y \mid \mathbf{x}, f\left(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i\right)\right) P(\boldsymbol{\theta}_i, \mathbf{m}_i) P(\mathbf{x}), \tag{7.4}$$

where $\mathbf{x}$ and $y$ represent samples and labels from $\mathcal{D}_i \cup \mathbf{M}_{i-1}$. The first term of the decomposition of Eq. 7.4 is the likelihood of correct labels given the input and the model prediction, while the joint distribution $P(\boldsymbol{\theta}_i, \mathbf{m}_i)$ describes the relation between model parameters $\boldsymbol{\theta}_i$ and the freezing strategy defined by $\mathbf{m}_i$. Assuming the independence between $\boldsymbol{\theta}_i$ and $\mathbf{m}_i$, this dis-

tribution can be expressed as:

$$P(\boldsymbol{\theta}_i, \mathbf{m}_i) = P(\boldsymbol{\theta}_i \mid \mathbf{m}_i)P(\mathbf{m}_i), \tag{7.5}$$

where

$$P(\boldsymbol{\theta}_i \mid \mathbf{m}_i) = \prod_j \mathcal{N}\left(\theta_{i,j}; \theta_{i-1,j}, \sigma_i^2\right)\Big)^{1-m_{i,j}}. \tag{7.6}$$

In this formulation, we model the distribution of each parameter $\theta_{i,j}$ as a Gaussian distribution depending on the corresponding mask value $m_{i,j}$, which removes a term from the overall probability when $m_{i,j} = 1$. Note that the mean of each parameter is set to $\theta_{i-1,j}$, i.e., its value at the end of the previous task (or to 0 for the first task, based on common initialization strategies).

In order to model $P(\mathbf{m}_i)$ in a practically feasible way, we employ some simplifying assumption based on the layered structure of deep learning models. Given $f = l_1 \circ l_2 \circ \cdots \circ l_L$, where each $l_k$ represents a network layer with parameters $\boldsymbol{\theta}_{|k}$ and $\boldsymbol{\theta} = \left[\boldsymbol{\theta}_{|1}, \ldots, \boldsymbol{\theta}_{|L}\right]$, let us similarly define $\mathbf{0}_{|k}$ and $\mathbf{1}_{|k}$ as two tensors with the same size as $\boldsymbol{\theta}_{|k}$, with all values set to 0 and 1, respectively. Then, we impose that possible values for $\mathbf{m}_i$ must be parameterized by a value $l$ as follows:

$$\mathbf{m}_i(l) = \left[\mathbf{1}_{|1}, \ldots, \mathbf{1}_{|l}, \mathbf{0}_{|l+1}, \ldots, \mathbf{0}_{|L}\right] \vee \mathbf{m}_{i-1} \tag{7.7}$$

with $l \in \{1, \ldots, L\}$. In practice, parameters frozen at previous tasks must remain so at the current task, and a layer's parameters can only be frozen altogether if all previous layers are also frozen.

Given these constraints, our goal is to find the optimal binary mask $\mathbf{m}_i$ that maximizes the likelihood of the labels $y$ given the inputs $\mathbf{x}$ from current task $\mathcal{D}_i$ and from long-term memory $\mathbf{M}_{i-1}$.This is expressed as the

following optimization problem:

$$\arg\max_{\mathbf{m}_i, \boldsymbol{\theta}_i} P\left(y \mid \mathbf{x}, f\left(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i\right)\right) P(\boldsymbol{\theta}_i \mid \mathbf{m}_i) P(\mathbf{m}_i) P(\mathbf{x}) \tag{7.8}$$

where the optimization is over parameters $\boldsymbol{\theta}_i$ and all feasible binary masks $\mathbf{m}_i$. Fast adaptation is thus carried out by maximizing this likelihood through the optimization of a loss function $\mathcal{L}$:

$$\mathcal{L}_{\text{fma}} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}\left[\mathcal{L}\left(y, f\left(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i\right)\right)\right] + \alpha \mathbb{E}_{(\mathbf{x},y)\sim\mathbf{M}_{i-1}}\left[\mathcal{L}\left(y, f\left(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i\right)\right)\right], \tag{7.9}$$

where $\mathbf{m}_i$ varies as described above, and $\alpha$ is a weighing factor between data sources. It is important to notice that, while optimizing for $\mathbf{m}_i$ necessarily requires updating $\boldsymbol{\theta}_i$ as well (since freezing, per se, does not alter inference performance), the objective is to prepare the model by identifying the optimal set of parameters that should be kept from previous tasks in a way that ensures both knowledge retainment and room for plasticity. For this reason, optimization is carried out for a single epoch over $\mathcal{D}_i$. Note that the choice of $\mathcal{L}$ is arbitrary: the proposed formulation allows for plugging in any existing continual learning method, enhancing it with the proposed training strategy.

### Sleep phase

During sleep, the brain cycles multiple times through two phases, known as rapid eye movement (REM) and non-rapid eye movement (NREM) sleep. In the NREM phase, the hippocampus replays and consolidates the information acquired at waking time by facilitating its transfer to the neocortex, where long-term memory storage occurs [203, 194]. REM sleep is thought to play a role in creativity and problem-solving [204, 205], allowing the brain to form new connections and generate novel ideas.

In our WSCL approach, we analogously distinguish between two alternating training modalities, conceptually mapped to the NREM and REM phases. During the former, we access examples from the current task (stored in the short-term memory) and from previous tasks (retrieved from long-term memory) to train the model — partially frozen during the wake stage — and stabilizing present knowledge. In the REM stage, we emulate the dreaming process by providing the model with examples from an external data source, with classes unrelated to any continual learning task. This approach allows the model to learn task-agnostic features which can be interpreted as a prior knowledge supporting task-specific learning and forward transfer.

### NREM stage

The main objective of this stage is to transfer information from the short-term memory $\mathbf{M}_s$, built in the precedent wake phase, to the model, strengthening the synaptic connections associated to the current task and thus enforcing plasticity, while retaining previously acquired knowledge thanks to long-term memory $\mathbf{M}_{i-1}$. In this setting, we apply parameter freezing mask $\mathbf{m}_i$ (defined in the wake phase), which is however not updated in the process.

Formally, in this stage we model the same distribution as in Eq. 7.4, but optimize for $\boldsymbol{\theta}_i$ alone, while leaving $\mathbf{m}_i$ constant. The objective thus becomes:

$$\arg\max_{\boldsymbol{\theta}_i} P\left(y \mid \mathbf{x}, f\left(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i\right)\right) P(\boldsymbol{\theta}_i \mid \mathbf{m}_i) P(\mathbf{x}), \tag{7.10}$$

where the prior on parameters $P\left(\boldsymbol{\theta}_i \mid \mathbf{m}_i\right)$ is essentially the same as in Eq. 7.6, with the difference that the mean of the distribution is the value of

$\boldsymbol{\theta}_i$ as computed at the end of the wake stage, rather than $\boldsymbol{\theta}_{i-1}$. Optimizing the above objective amounts to minimizing a variant of the loss in Eq. 7.9:

$$\mathcal{L}_{\text{NREM}} = \mathbb{E}_{(\mathbf{x},y)\sim\mathbf{M}_s}\left[\mathcal{L}\left(y, f\left(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i\right)\right)\right] + \alpha\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{M}_{i-1}}\left[\mathcal{L}\left(y, f\left(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i\right)\right)\right], \tag{7.11}$$

where $\mathbf{M}_s$ is employed instead of the whole dataset $\mathcal{D}_i$.

In this stage, we also gradually update long-term memory $\mathbf{M}_i$, using reservoir sampling [17] to inject task experience from short-term memory $\mathbf{M}_s$ into $\mathbf{M}_i$, so that it becomes available to future tasks.

### REM stage

We approximate the sleeping mechanism performed by the human brain in the REM stage by providing the model with an additional source of previously unseen knowledge (a "dreaming" dataset with no semantic overlap with CL classes), that can help the model to generalize better to new and unseen data, as suggested by cognitive literature [202].

Let $\mathcal{D}_{\text{dream}}$ be the dreaming dataset from which we can sample data points $(\mathbf{x}, y) \sim \mathcal{D}_{\text{dream}}$, with $\mathbf{x} \in \mathcal{X}$ and class label $y \in \mathcal{Y}_{\text{dream}}$. We assume that $\mathcal{Y}_{\text{dream}} \cap \mathcal{Y} = \emptyset$ (the latter being the set of continual learning classes), to prevent any overlap between auxiliary and continual learning classes. Given this premise, the proposed optimization objective becomes:

$$\arg\max_{\boldsymbol{\theta}_i} P\left(y \mid \mathbf{x}, f\left(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i\right)\right) P(\boldsymbol{\theta}_i \mid \mathbf{m}_i) P(\mathbf{x}), \tag{7.12}$$

where $(\mathbf{x}, y) \sim \mathcal{D}_{\text{dream}}$, while the other terms are the same as in Eq. 7.10. This objective is then mapped to a training loss function defined as:

$$\mathcal{L}_{\text{REM}} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{\text{dream}}}\left[\mathcal{L}\left(y, f\left(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i\right)\right)\right]. \tag{7.13}$$

During REM stage, training with two distinct class label sets, $\mathcal{Y}$ from the continual learning problem and $\mathcal{Y}_{\text{dream}}$ from the dreaming dataset has been addressed following the procedure reported in [111].

## 7.4 Experimental Evaluation

### 7.4.1 Benchmarks

We test WSCL on several continual learning benchmarks obtained by taking image classification datasets and splitting their classes equally into a series of disjoint tasks. Moreover, since REM stage requires additional dreaming samples, for each benchmark we also identify its dreamed-counterpart:

- **Seq-CIFAR-10** [25], a widely-used image classification dataset obtained by splitting CIFAR-10 images into 5 binary classification tasks. Its counterpart used for the REM stage consists of a subset of 50 CIFAR-100 classes, selected after removing those with semantic relations to CIFAR-10.
- **Seq-FG-ImageNet**[2] is a fine-grained image classification benchmark with 100 classes of animals, used to test CL methods on a more challenging task. The dreaming counterpart consists of additional 100 classes taken from ImageNet, after removing all synsets derived from "organism".
- **Tiny-ImageNet** [15] is a subset of ImageNet consisting of 200 classes with 500 images each, resized to 64×64. We employ the first 100 classes as the main training dataset **Seq-Tiny-ImageNet$_{1/2}$** (organized as 5 tasks

[2]Seq-FG-ImageNet is derived from https://www.kaggle.com/datasets/ambityga/imagenet100

of 20 classes) and the remaining 100 classes as the dreaming dataset.

### 7.4.2 Training Procedure

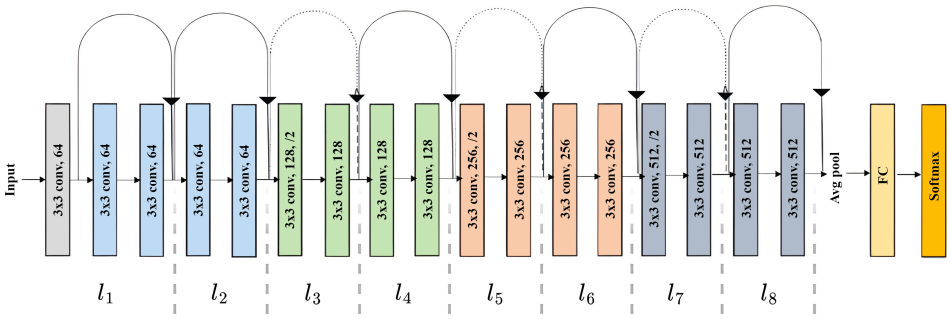



***Figure 7.2:*** *ResNet-18 architecture and selective freezing strategy. After an initial convolutional layer (apart from the last fully-connected layer), each colored portion represents a network's* main block*, consisting of two residual* basic blocks*, each applying two convolutions. For our freezing strategy, we treat each basic block as the smallest unit of freezing, named* layer.

Our approach employs a ResNet-18 backbone for feature extraction and classification. As showed in Fig. 7.2, ResNet-18 includes, at a high level, four *main blocks* (depicted with distinct colors in the figure), each of which includes two *basic blocks*[3]; each basic block applies two convolutions with a residual connection. With reference to the definition of model $f$ in Sect. 7.3.1, we will treat each block as the smallest unit of freezing,

[3] https://pytorch.org/vision/master/_modules/torchvision/models/resnet.html

named *layer*. Consequently, the network is divided in $L = 8$ *layers*. The first convolution of the network is assumed to be part of the first layer.

In the wake stage of task $i$, we train multiple instances of the model, starting from parameters $\boldsymbol{\theta}_i$, with all possible configurations of $\mathbf{m}_i$: if the deepest frozen layer is $l_j$, the number of possible values for $\mathbf{m}_i$ is $L-j+1$, with $L$ being the total number of layers. Training is carried out for a single epoch with mini-batch SGD and a learning rate of 0.03. Batch size is set to 32 for CIFAR-10 and Tiny-ImageNet$_{1/2}$, and to 8 for FG-ImageNet. The $\alpha$ hyperparameter in Eq. 7.9 is set to 1, and the $N_s$ dimension of the short-term buffer to 5,000. It is important to mention that, in our implementation, the optimization of Eq. 7.9 (fast model adaptation loss $\mathcal{L}_{\text{fma}}$) and Eq. 7.11 (NREM loss $\mathcal{L}_{\text{NREM}}$) on long-term memory $\mathbf{M}_i$ is carried out on disjoint portions of the whole set of stored samples. In particular, 10% of $\mathbf{M}_i$ is used when optimizing $\mathcal{L}_{\text{fma}}$, while the remaining 90% is used for $\mathcal{L}_{\text{NREM}}$. This separation mitigates the risk of overfitting of $\mathcal{L}_{\text{NREM}}$ on data that will be used, in the wake phase, to determine to which extent model layers should be frozen: indeed, in case of overfitting, the wake phase would encourage model freezing, as it would more easily minimize the corresponding loss term.
In the sleep stage, we train the model using $\mathcal{L}_{\text{NREM}}$ and the $\mathcal{L}_{\text{REM}}$ losses at alternately batches. We perform 10 epochs of training, with the same optimizer settings and hyperparameters as above.

### 7.4.3 Results

We first evaluate how WSCL contributes to classification accuracy of state-of-the-art models. To accomplish this, we select recent rehearsal-based methods, namely, DER++ [38], ER-ACE [40] and ER [47], and compare

their performance when the WSCL training strategy is employed, by plugging them in as the $\mathcal{L}$ loss term in Eq. 7.9, 7.11, 7.13. We address rehearsal-based methods only, as WSCL requires a memory buffer to model long-term memory. We report *final average accuracy (FAA)* after training on the last task in the Class-Incremental and Task-Incremental settings.

We further provide a lower bound, consisting of training without any countermeasure to forgetting (*Fine-tune*), and an upper bound given by training all tasks jointly (*Joint*). Results in Table 7.1 show that, on all three benchmarks, WSCL leads to a significant performance gain that varies from about 2 percent points on FG-ImageNet to 12 percent points on CIFAR-10, substantiating our claims on the importance of leveraging human learning strategies for building better computational methods. Table 7.1 also reports the comparison with: a) DualNet [43], which leverages CLS theory and the same backbone, i.e., ResNet-18; b) CoPE [121] that integrates contrastive learning — another technique inspired by cognitive neuroscience [202] — for better feature transferability to later tasks[4]. We do not include DualPrompt [193] as it uses a large pre-trained ViT [206] as a backbone, leading to an unfair comparison with the simpler ResNet-18. All methods combined with our WSCL strategy improve over DualNet (up to about 40 percent points) and CoPE, demonstrating how mimicking human learning more strictly improves performance even in a purely discriminative supervised learning regime. We also measure *forward transfer (FWT)*, a desirable property in CL that indicates how much a model leverages previous knowledge for learning a new task [17]. Forward transfer is estimated as the average difference between a task's accuracy when learning it in a CL setting and when learning it from random initializa-

[4]Results for DualNet and CoPE are computed using their original implementations and hyperparameters.

| **Method** | **Seq-CIFAR-10** | | **Seq-Tiny-ImageNet$_{1/2}$** | | **Seq-FG-ImageNet** | |
|---|---|---|---|---|---|---|
| | *Class-IL* | | | | | |
| Joint | 85.15±1.99 | | 50.81±1.65 | | 43.39±1.76 | |
| Fine-tune | 19.47±0.10 | | 13.84±0.55 | | 3.88±0.33 | |
| **Buffer size** | **200** | **500** | **200** | **500** | **200** | **1000** |
| ER [40] | 48.76±0.57 | 59.75±2.51 | 16.25±0.85 | 21.07±1.43 | 4.23±0.15 | 5.05±0.51 |
| ↪**WSCL** | 51.86±4.40 | 63.71±1.35 | 18.81±0.48 | 23.63±0.95 | 6.01±0.64 | 15.26±3.39 |
| DER++ [38] | 57.35±5.47 | 69.06±1.24 | 16.62±1.76 | 23.40±1.66 | 5.95±0.49 | 8.59±1.11 |
| ↪**WSCL** | 63.97±3.38 | 72.33±0.99 | 23.70±0.91 | 31.81±0.70 | 6.48±1.22 | 11.70±0.14 |
| ER-ACE [47] | 59.98±2.65 | 67.17±1.54 | 27.81±1.24 | 32.10±2.21 | 9.42±0.78 | 11.58±3.59 |
| ↪**WSCL** | 71.15±2.15 | 74.18±1.28 | 35.68±1.18 | 41.25±1.75 | 12.51±0.86 | 20.51±0.56 |
| | *Task-IL* | | | | | |
| Joint | 96.90±0.14 | | 71.50±1.31 | | 85.47±1.56 | |
| Fine-tune | 66.53±5.66 | | 33.87±1.39 | | 30.33±3.12 | |
| **Buffer size** | **200** | **500** | **200** | **500** | **200** | **1000** |
| ER [40] | 90.88±0.69 | 91.88±1.43 | 48.79±1.51 | 57.49±1.87 | 50.85±1.43 | 59.33±0.70 |
| ↪**WSCL** | 92.32±0.70 | 94.43±0.18 | 57.13±0.97 | 61.96±0.91 | 56.47±0.49 | 69.96±2.52 |
| DER++ [38] | 90.53±1.52 | 92.98±0.37 | 51.31±2.17 | 59.30±2.06 | 51.45±4.45 | 65.25±0.86 |
| ↪**WSCL** | 93.38±1.12 | 94.28±0.46 | 61.48±0.78 | 67.23±1.01 | 49.21±4.14 | 57.00±1.73 |
| ER-ACE [47] | 91.81±0.31 | 92.96±0.33 | 53.00±1.86 | 57.35±2.00 | 57.32±1.80 | 60.00±6.16 |
| ↪**WSCL** | 94.78±0.75 | 94.96±0.49 | 59.82±1.10 | 65.38±1.75 | 56.90±2.63 | 67.39±1.24 |

***Table 7.1:*** *Final average accuracy (FAA) [↑] of rehearsal-based methods, with and without WSCL, for different buffer sizes.*

| **FWT** | **Seq-CIFAR-10** | | **Seq-Tiny-ImageNet$_{1/2}$** | | **Seq-FG-ImageNet** | |
|---|---|---|---|---|---|---|
| Joint | 85.15 | | 50.81 | | 43.39 | |
| Fine-tune | 19.47 | | 13.84 | | 3.88 | |
| **Buffer size** | **200** | **500** | **200** | **500** | **200** | **1000** |
| ER [40] | −7.36 | −12.20 | −1.00 | −1.32 | −1.05 | −1.02 |
| ↪**WSCL** | 1.68 | 6.03 | 12.41 | 12.60 | 3.82 | 3.17 |
| DER++ [38] | −12.29 | −6.23 | −0.84 | −1.06 | −0.08 | −1.05 |
| ↪**WSCL** | 1.06 | 2.83 | 12.16 | 12.24 | 1.78 | 2.31 |
| ER-ACE [47] | −8.58 | −8.97 | −0.73 | −0.94 | −1.04 | −1.17 |
| ↪**WSCL** | 0.48 | −1.87 | 8.60 | 9.06 | 1.83 | 1.19 |
| **DualNet** [43] | −5.89 | −7.41 | −0.78 | −0.96 | −1.03 | −1.96 |
| **CoPE** [44] | −3.63 | −4.23 | −0.87 | −1.05 | −0.98 | −1.23 |

***Table 7.2:*** *Forward Transfer (FWT) of rehearsal-based methods, with and without WSCL, for different buffer sizes.*

tion (details in [17]). Table 7.2 shows how WSCL tends to enhance FWT, bringing it from negative to positive values. This is highly remarkable as the majority of existing CL methods show a negative forward transfer. It is equally important to measure forgetting (the lower, the better) to assess how well an approach tackles *no-i.i.d.* data. Cross-checking results in Table 7.3 with those in Tables 7.1 and 7.2 highlights how WSCL effectively reduces forgetting while enhancing Forward Transfer skills and accuracy performance in a way sensibly higher than the baselines. Therefore, this set of experiments underscores the capabilities of the WSCL strategy for reducing forgetting and preparing the network for future tasks.

| **Method** | **Seq-CIFAR-10** | | **Seq-Tiny-ImageNet$_{1/2}$** | | **Seq-FG-ImageNet** | |
|---|---|---|---|---|---|---|
| **Buffer size** | **200** | **500** | **200** | **500** | **200** | **1000** |
| ER [40] | 56.66 | 43.21 | 62.63 | 58.16 | 74.04 | 73.45 |
| ↪**WSCL** | 50.23 | 36.04 | 56.71 | 50.63 | 76.79 | 63.93 |
| DER++ [38] | 31.23 | 22.63 | 62.15 | 50.81 | 67.10 | 63.63 |
| ↪**WSCL** | 35.53 | 23.52 | 51.30 | 43.91 | 59.84 | 52.39 |
| ER-ACE [47] | 16.55 | 15.21 | 34.41 | 28.15 | 32.61 | 36.44 |
| ↪**WSCL** | 11.78 | 10.69 | 28.23 | 23.29 | 27.24 | 33.53 |

***Table 7.3:*** *Forgetting of rehearsal-based methods, with and without WSCL, for different buffer sizes, in Class-IL setting.*

### 7.4.4 Model Analysis

Model analysis is mainly carried out using ER-ACE (the best-performing method from Table 7.1) as baseline, on the Tiny-ImageNet$_{1/2}$ dataset. We first ablate the processing phases of WSCL: results in Table 7.4 show how the NREM/REM sleep states equally contribute to the final model performance. Interestingly, the REM phase is responsible for positive forward transfer, which is consistent with cognitive neuroscience evidence that REM prepares brain synapses to future experience [204, 205].

We then evaluate the impact of the quality of dreaming, by adding Gaussian noise (at different percentages) and reducing the spatial resolution of dreaming samples. Fig. 7.3 indicates that WSCL still outperforms the baseline when dreaming images are affected by noise up to 30% or scaled down by $6\times$, suggesting that the role of REM stage in consolidating knowledge is mostly independent from the visual details of the dreamed

| Method | FAA | FWT |
|---|---|---|
| Only Wake | 4.70 | -0.93 |
| Wake + REM | 25.68 | 11.89 |
| Wake + NREM | 27.61 | -0.67 |
| Wake + REM + NREM | 35.68 | 8.60 |

***Table 7.4:*** *Ablation on the WSCL processing stages: results refer to ER-ACE on Tiny-ImageNet$_{1/2}$.*

samples, which merely serve to learn additional reusable features.

Besides the quality, the quantity may also be a crucial issue. Here we further investigate the impact of the size of the dreaming dataset in the results. Figure 7.4 illustrates how the dreaming stage allows for enhanced performance even when the additional dreaming dataset is reduced by approximately 70%.

We finally assess the efficiency aspects of WSCL. Indeed, the human brain is capable of performing complex tasks with remarkable speed and accuracy, at a relatively low energy cost: cerebral parallel processing architecture, plasticity, and ability to adapt to changing environments are all factors that contribute to its efficiency [207, 208]. In WSCL, efficiency is encouraged in the wake stage, by letting the model selectively freeze different portions of the network: this is analogous and consistent to cognitive neuroscience evidence that a synchronization of neural activity across different brain regions and changes in the balance between excitation and inhibition enable efficient processing [209, 210].

Fig. 7.5 shows the most frequent (over 10 different runs) set of frozen backbone layers at each task, when training ER-ACE with WSCL on Tiny-

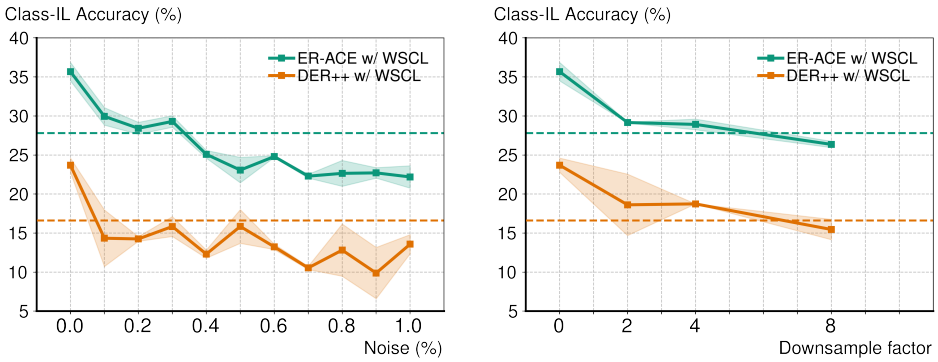



***Figure 7.3:*** *Impact of dreaming quality, in terms of noise (left) and image resolution (right). Results refer to ER-ACE and DER++ with WSCL (solid lines) and without it (dotted line).*

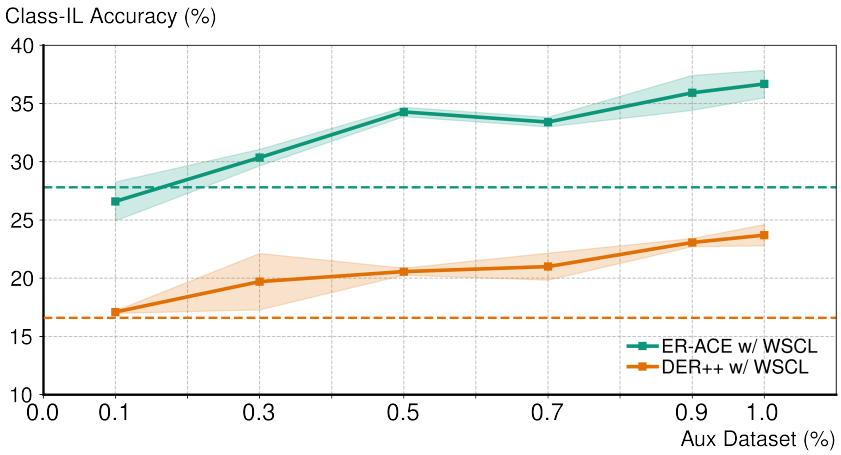



***Figure 7.4:*** *Impact of dreaming dataset dimension. Results refer to ER-ACE and DER++ with WSCL (solid lines) and without it (dotted line).*

ImageNet$_{1/2}$, as well as the total number of performed parameter updates using the training procedure presented in Sect. 7.4.2. WSCL's training

procedure reduces the overall number of updates for the entire training of the ResNet-18 model, by a quantity that tends to increase with the number of training epochs (from 2% to about 17% less updates), thus confirming the suitability of the wake stage in supporting efficient training.

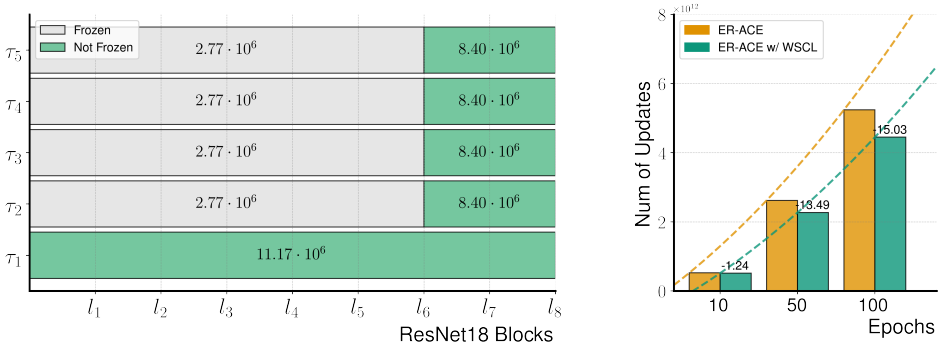



***Figure 7.5: WSCL model efficiency****: Left: the most frequent automatically learned freezing scheme (values within bars are number of parameters) during the wake phase for ER-ACE on Tiny-ImageNet$_{1/2}$. Right: number of parameter updates for the whole training of ER-ACE with and without WSCL on Tiny-ImageNet$_{1/2}$ (from 10 epochs to 100 training epochs).*

## 7.5 Discussion

The integration of Complementary Learning Systems (CLS) theory and sleep mechanisms in artificial neural networks holds great potential for enhancing continual learning capabilities. Inspired by the interaction between the hippocampus and neocortex in humans, Wake-Sleep Consolidated Learning (WSCL) introduces a sleep phase that mimics off-line brain states during which memory consolidation and synaptic reorganization oc-

cur. By leveraging the wake phase for fast adaptation and episodic memory formation, and the sleep phase for memory consolidation and dreaming, WSCL shows superior performance compared to prior work on various benchmarks. Importantly, WSCL achieves positive forward transfer, exhibiting the ability to prepare synapses for future knowledge. These findings highlight the importance of all three stages — wake, NREM and REM — in supporting network plasticity and reducing forgetting for improved learning and memory.

Future research will address the advancement of memory and dreaming modeling techniques, which currently rely on conventional rehearsal methods to facilitate memory retention and on the employment of external datasets for generating dream-like experiences. With regard to memory modeling, it is essential to delve into more nuanced and dynamic approaches that accurately capture the intricacies of memory formation, storage, and retrieval, by also devising mechanisms to account for memory decay and interference. Likewise, for dream modeling, there is an opportunity to push beyond the current reliance on external datasets and explore more sophisticated techniques. This could entail developing generative models capable of simulating dream-like experiences based on the network's existing knowledge and latent representations. By accomplishing this, the model's ability to generate diverse, creative, and contextually relevant dream scenarios can be elevated to a new level of realism.

It is important to acknowledge that, while the pursuit of more realistic memory and dreaming modeling techniques is desirable, their integration into the WSCL framework is possible thanks to its modular architecture, which provides a solid foundation that can accommodate the inclusion of advanced components dedicated to specific aspects of memory management or sample generation.

## 7.6 Publications

The approach described in this chapter has led to two publications.
An initial work investigating the possibility of selectively freezing parts of the backbone network while maintaining competitive performance and reducing computational and energy requirements was presented at the 2023 International Conference on Computer Vision (ICCV), Visual Continual Learning Workshop, Paris, France:

- Sorrenti, A. Bellitto, G., Proietto Salanitri, F., Pennisi, M., Spampinato, C., Palazzo, S. (2023). Selective Freezing for Efficient Continual Learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3550-3559).

A second paper presenting the proposed WSCL framework is currently under review at IEEE Transactions on Neural Networks and Learning Systems (TNNLS) journal:

- Sorrenti, A., Bellitto, G., Proietto Salanitri, F., Pennisi, M., Palazzo, S., Spampinato, C. (2023). Wake-Sleep Consolidated Learning. Submitted to IEEE Transactions on Neural Networks and Learning Systems.

# Part IV

# CONCLUSIONS

In this thesis we delved into the study of classical problems encountered in Continual Learning from a new perspective, i.e., the one of emulating human cognitive mechanisms. As the AI researchers push the boundaries of what neural networks can achieve, the complexity increases and so do the challenges. Continual Learning, once a peripheral task of the computer vision community, is now at the center of these challenges, especially because of the problem of catastrophic forgetting. It currently represents an almost insurmountable barrier to developing the next generation of intelligent agents equipped with effective incremental learning capabilities.

The aim of this dissertation was to provide new contributions, inspired by *how* humans learn, which might enable the community to take a step forward in this direction.

After a high-level introduction to the Catastrophic Forgetting problem discussed in Chapter 1, in Chapter 2 we delved into the formal definition of the task of Continual Learning for image classification, analyzing its main challenges, the benchmarks used, and giving a brief overview of the state of the art.

The innate human ability to value past experiences for enhancing future learning lays the foundation for next chapters.

In Chapter 3 we highlighted the importance of using past experience, elucidating how previous encounters can improve problem solving. In this first approach, knowledge is replicated via an auxiliary data stream that enhances the model's pattern recognition and knowledge generalization during training, allowing it to quickly adapt to new tasks.

In Chapter 4, while we aimed to mimic human cognitive processes by utilizing pre-trained models, we face the inherent limitations of transfer learning within the CL structure, finding that pre-training suffers from forgetting as well. Then, in Chapter 5, we proposed to merge the CL classification with a supplementary task that steering the learning process, with a pronounced focus on the integration of self-supervised equivariant tasks.

While these chapters focused on solutions to replicate prior knowledge in neural networks, in later chapters we shift our effort towards strategies that attempt to emulate the learning process, drawing broadly on neurocognitive theories of human learning.

In Chapter 6 we found that selective attention, encoded as a saliency prediction task, is robust to catastrophic forgetting, and that saliency features can improve a CL model, while in Chapter 7 we proposed a method that emulates the wake-sleep alternation, which in humans is fundamental to consolidate memories and facilitating learning, by increasing generalization of knowledge.

While we are still a long way from creating machines that can learn as seamlessly as humans, the journey has begun with renewed vigor and perspective. As we continue to learn from the complexities and wonders of human learning, the solutions to the CL problems will hopefully become clearer, paving the way for an era in which machines can learn, adapt, and

evolve as effortlessly as humans do.

# BIBLIOGRAPHY

[1] G. Carpenter and S. Grossberg, "Adaptive resonance theory: Stable self-organization of neural recognition codes in response to arbitrary lists of input patterns," in *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pp. 45–62, Erlbaum, 1986.

[2] G. E. Hinton and D. C. Plaut, "Using fast weights to deblur old memories," in *Proceedings of the ninth annual conference of the Cognitive Science Society*, pp. 177–186, 1987.

[3] R. S. Sutton, "Two problems with backpropagation and other steepest-descent learning procedures for networks," in *Proc. of Eightth Annual Conference of the Cognitive Science Society*, pp. 823–831, 1986.

[4] J. L. McClelland, D. E. Rumelhart, P. R. Group, *et al.*, *Parallel distributed processing, volume 2: Explorations in the microstructure of cognition: Psychological and biological models*, vol. 2. MIT press, 1987.

[5] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of learning and motivation*, 1989.

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

[7] S. Grossberg, *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition and Motor Control*. Springer Science & Business Media, 1982.

[8] G. M. van de Ven and A. S. Tolias, "Three continual learning scenarios," in *Neural Information Processing Systems Workshops*, 2018.

[9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, 2009.

[10] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE signal processing magazine*, vol. 32, no. 3, pp. 53–69, 2015.

[11] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[12] M. J. Mirza, M. Masana, H. Possegger, and H. Bischof, "An efficient domain-incremental learning approach to drive in all weather conditions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3001–3011, 2022.

[13] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," tech. rep., Citeseer, 2009.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, 2015.

[15] Stanford, "Tiny ImageNet Challenge (CS231n)," 2015. https://www.kaggle.com/c/tiny-imagenet.

[16] C. Wah, S. Branson, P. Welinder, P. Perona, and S. J. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

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

[18] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[19] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Neural Information Processing Systems Workshops*, 2015.

[20] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

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

[22] A. Rannen, R. Aljundi, M. B. Blaschko, and T. Tuytelaars, "Encoder based lifelong learning," in *Proceedings of the IEEE international conference on computer vision*, pp. 1320–1328, 2017.

[23] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, 2017.

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

[25] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning*, 2017.

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

[27] A. Mallya and S. 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.

[28] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *International Conference on Machine Learning*, 2018.

[29] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Advances in Neural Information Processing Systems*, 2017.

[30] Y. Xiang, Y. Fu, P. Ji, and H. Huang, "Incremental learning using conditional adversarial networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6619–6628, 2019.

[31] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[32] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Science*, 1995.

[33] J. S. Vitter, "Random sampling with a reservoir," *ACM Transactions on Mathematical Software*, 1985.

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

[35] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient Lifelong Learning with A-GEM," in *International Conference on Learning Representations*, 2019.

[36] A. S. Benjamin, D. Rolnick, and K. Kording, "Measuring and regularizing networks in function space," in *International Conference on Learning Representations*, 2019.

[37] A. Chaudhry, A. Gordo, P. K. Dokania, P. Torr, and D. Lopez-Paz, "Using hindsight to anchor past knowledge in continual learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[38] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark Experience for General Continual Learning: a Strong, Simple Baseline," in *Advances in Neural Information Processing Systems*, 2020.

[39] M. Boschini, P. Buzzega, L. Bonicelli, A. Porrello, and S. Calderara, "Continual semi-supervised learning through contrastive interpolation consistency," *Pattern Recognition Letters*, vol. 162, pp. 9–14, 2022.

[40] L. Caccia, R. Aljundi, N. Asadi, T. Tuytelaars, J. Pineau, and E. Belilovsky, "New Insights on Reducing Abrupt Representation Change in Online Continual Learning," in *International Conference on Learning Representations*, 2022.

[41] H. Cha, J. Lee, and J. Shin, "Co2l: Contrastive continual learning," in *IEEE International Conference on Computer Vision*, 2021.

[42] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised Contrastive Learning," in *Advances in Neural Information Processing Systems*, 2020.

[43] Q. Pham, C. Liu, and S. Hoi, "Dualnet: Continual learning, fast and slow," in *Advances in Neural Information Processing Systems*, 2021.

[44] M. D. Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. G. Slabaugh, and T. Tuytelaars, "Continual learning: A comparative study on how to defy forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[45] J. Cichon and W.-B. Gan, "Branch-specific dendritic ca 2+ spikes cause persistent synaptic plasticity," *Nature*, 2015.

[46] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, 2019.

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

[48] R. M. French, "Using semi-distributed representations to overcome catastrophic forgetting in connectionist networks," in *Proceedings of the Annual Conference of the Cognitive Science Society*, 1991.

[49] K. McRae and P. A. Hetherington, "Catastrophic interference is eliminated in pretrained networks," in *Proceedings of the Annual Conference of the Cognitive Science Society*, 1993.

[50] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *International Conference on Machine Learning*, 2019.

[51] S. Sakellaridi, V. Christopoulos, T. Aflalo, K. Pejsa, E. Rosario, D. Ouellette, N. Pouratian, and R. Andersen, "Intrinsic variable learning for brain-machine interface control by human anterior intraparietal cortex," 2019.

[52] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, 2016.

[53] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, 2019.

[54] S. Lewandowsky and S.-C. Li, "Catastrophic interference in neural networks: Causes, solutions, and data," in *Interference and inhibition in cognition*, Elsevier, 1995.

[55] A. Chaudhry, N. Khan, P. K. Dokania, and P. H. Torr, "Continual learning in low-rank orthogonal subspaces," in *Advances in Neural Information Processing Systems*, 2020.

[56] S. Jung, H. Ahn, S. Cha, and T. Moon, "Continual learning with node-importance based adaptive group sparse regularization," in *Advances in Neural Information Processing Systems*, 2020.

[57] J. Rajasegaran, M. Hayat, S. H. Khan, F. S. Khan, and L. Shao, "Random path selection for continual learning," in *Advances in Neural Information Processing Systems*, 2019.

[58] I. J. Goodfellow, M. Mirza, X. Da, A. C. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgeting in gradient-based neural networks," *International Conference on Learning Representations*, 2014.

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

[60] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient Lifelong Learning with A-GEM," in *International Conference on Learning Representations*, 2019.

[61] M. Farajtabar, N. Azizan, A. Mott, and A. Li, "Orthogonal gradient descent for continual learning," in *International Conference on Artificial Intelligence and Statistics*, 2020.

[62] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," in *Advances in Neural Information Processing Systems*, 2019.

[63] M. Riemer, T. Klinger, D. Bouneffouf, and M. Franceschini, "Scalable recollections for continual lifelong learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

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

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

[66] C. D. Kim, J. Jeong, S. Moon, and G. Kim, "Continual learning on noisy data streams via self-purified replay," in *IEEE International Conference on Computer Vision*, 2021.

[67] A. Prabhu, P. H. Torr, and P. K. Dokania, "GDumb: A simple approach that questions our progress in continual learning," in *Proceedings of the European Conference on Computer Vision*, 2020.

[68] S. Farquhar and Y. Gal, "Towards Robust Evaluations of Continual Learning," in *International Conference on Machine Learning Workshop*, 2018.

[69] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *International Conference on Pattern Recognition*, 2016.

[70] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference," in *International Conference on Learning Representations*, 2019.

[71] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014.

[72] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training

for high fidelity natural image synthesis," in *International Conference on Learning Representations*, 2019.

[73] G. Rothschild, E. Eban, and L. M. Frank, "A cortical-hippocampal-cortical loop of information processing during memory consolidation," 2017.

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision*, 2015.

[75] A. Porrello, S. Vincenzi, P. Buzzega, S. Calderara, A. Conte, C. Ippoliti, L. Candeloro, A. Di Lorenzo, and A. C. Dondona, "Spotting insects from satellites: modeling the presence of culicoides imicola through deep cnns," in *International Conference on Signal-Image Technology & Internet-Based Systems*, 2019.

[76] S. Allegretti, F. Bolelli, F. Pollastri, S. Longhitano, G. Pellacani, and C. Grana, "Supporting Skin Lesion Diagnosis with Content-Based Image Retrieval," in *International Conference on Pattern Recognition*, 2021.

[77] K. Shaheen, M. A. Hanif, O. Hasan, and M. Shafique, "Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks," *Journal of Intelligent & Robotic Systems*, 2022.

[78] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015.

[79] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *IEEE International Conference on Computer Vision*, 2017.

[80] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[81] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*, 2020.

[82] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *International Conference on Machine Learning*, 2017.

[83] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, 2018.

[84] S. V. Mehta, D. Patil, S. Chandar, and E. Strubell, "An empirical investigation of the role of pre-training in lifelong learning," in *International Conference on Machine Learning*, 2021.

[85] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proceedings of the European Conference on Computer Vision*, 2018.

[86] M. Boschini, L. Bonicelli, P. Buzzega, A. Porrello, and S. Calderara, "Class-incremental continual learning into the extended der-verse,"

*IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5497–5512, 2022.

[87] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, 2017.

[88] A. Nichol and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[89] P. Buzzega, M. Boschini, A. Porrello, and S. Calderara, "Rethinking Experience Replay: a Bag of Tricks for Continual Learning," in *International Conference on Pattern Recognition*, 2020.

[90] J. Smith, J. Balloch, Y.-C. Hsu, and Z. Kira, "Memory-efficient semi-supervised continual learning: The world is its own replay buffer," in *International Joint Conference on Neural Networks*, 2021.

[91] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, 2014.

[92] Y. Jang, H. Lee, S. J. Hwang, and J. Shin, "Learning what and where to transfer," in *International Conference on Machine Learning*, 2019.

[93] K. Wang, X. Gao, Y. Zhao, X. Li, D. Dou, and C.-Z. Xu, "Pay attention to features, transfer learn faster cnns," in *International Conference on Learning Representations*, 2019.

[94] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[95] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *International Conference on Machine Learning*, 2019.

[96] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *International Conference on Learning Representations*, 2015.

[97] G. Aguilar, Y. Ling, Y. Zhang, B. Yao, X. Fan, and C. Guo, "Knowledge distillation from internal representations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[98] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *IEEE International Conference on Computer Vision*, 2019.

[99] A. Monti, A. Porrello, S. Calderara, P. Coscia, L. Ballan, and R. Cucchiara, "How many observations are enough? knowledge distillation for trajectory forecasting," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[100] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "Bam: Bottleneck attention module," in *British Machine Vision Conference*, 2018.

[101] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with

gumbel-softmax," in *International Conference on Learning Representations*, 2017.

[102] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *International Conference on Machine Learning*, 2018.

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

[104] R. Müller, S. Kornblith, and G. Hinton, "Subclass distillation," *arXiv preprint arXiv:2002.03936*, 2020.

[105] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, "Modulating early visual processing by language," in *Advances in Neural Information Processing Systems*, 2017.

[106] A. H. Robinson and C. Cherry, "Results of a prototype television bandwidth compression scheme," in *Proceedings of the IEEE*, 1967.

[107] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on information theory*, 1977.

[108] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2009.

[109] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, and J. v. d. Weijer, "Semantic drift compensation for class-incremental learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[110] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in *International Conference on Machine Learning*, 2018.

[111] G. Bellitto, M. Pennisi, S. Palazzo, L. Bonicelli, M. Boschini, and S. Calderara, "Effects of auxiliary knowledge on continual learning," in *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 1357–1363, IEEE, 2022.

[112] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Advances in Neural Information Processing Systems*, 2011.

[113] R. Dangovski, L. Jing, C. Loh, S. Han, A. Srivastava, B. Cheung, P. Agrawal, and M. Soljačić, "Equivariant contrastive learning," *arXiv preprint arXiv:2111.00899*, 2021.

[114] X. Chen and K. He, "Exploring simple siamese representation learning.," tech. rep., Facebook AI Research, 2020.

[115] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *International Conference on Machine Learning*, 2021.

[116] A. Bardes, J. Ponce, and Y. LeCun, "Vicreg: Variance-invariance-

covariance regularization for self-supervised learning," in *International Conference on Learning Representations*, 2022.

[117] E. Fini, V. G. T. da Costa, X. Alameda-Pineda, E. Ricci, K. Alahari, and J. Mairal, "Self-supervised models are continual learners," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[118] D. Madaan, J. Yoon, Y. Li, Y. Liu, and S. J. Hwang, "Rethinking the representational continuity: Towards unsupervised continual learning," in *International Conference on Learning Representations*, 2022.

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

[120] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online continual learning in image classification: An empirical survey," 2022.

[121] M. De Lange and T. Tuytelaars, "Continual prototype evolution: Learning online from non-stationary data streams," in *IEEE International Conference on Computer Vision*, 2021.

[122] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations*, 2018.

[123] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord, "Boosting few-shot visual learning with self-supervision," in *IEEE International Conference on Computer Vision*, 2019.

[124] R. Dangovski, L. Jing, C. Loh, S. Han, A. Srivastava, B. Cheung, P. Agrawal, and M. Soljačić, "Equivariant contrastive learning," in *International Conference on Learning Representations*, 2022.

[125] S. Addepalli, K. Bhogale, P. Dey, and R. V. Babu, "Towards efficient and effective self-supervised learning of visual representations," in *Proceedings of the European Conference on Computer Vision*, 2022.

[126] A. Chaudhry, A. Gordo, P. Dokania, P. Torr, and D. Lopez-Paz, "Using hindsight to anchor past knowledge in continual learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[127] V. Araujo, J. Hurtado, A. Soto, and M.-F. Moens, "Entropy-based stability-plasticity for lifelong learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[128] G. Lin, H. Chu, and H. Lai, "Towards better plasticity-stability trade-off in incremental learning: A simple linear connector," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[129] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proceedings of the European Conference on Computer Vision*, 2016.

[130] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell, and M. Rohrbach, "Adversarial continual learning," in *Proceedings of the European Conference on Computer Vision*, 2020.

[131] M. M. Derakhshani, X. Zhen, L. Shao, and C. Snoek, "Kernel continual learning," in *International Conference on Machine Learning*, 2021.

[132] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, "Importance estimation for neural network pruning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[133] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[134] Q. Gao, C. Zhao, B. Ghanem, and J. Zhang, "R-DFCIL: Relation-Guided Representation Learning for Data-Free Class Incremental Learning," in *Proceedings of the European Conference on Computer Vision*, 2022.

[135] Y. Gu, X. Yang, K. Wei, and C. Deng, "Not just selection, but exploration: Online class-incremental continual learning via dual view consistency," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[136] Y. Zhang, B. Pfahringer, E. Frank, A. Bifet, N. J. S. Lim, and Y. Jia, "A simple but strong baseline for online continual learning: Re-

peated Augmented Rehearsal," in *Advances in Neural Information Processing Systems*, 2022.

[137] J. Smith, Y.-C. Hsu, J. Balloch, Y. Shen, H. Jin, and Z. Kira, "Always be dreaming: A new approach for data-free class-incremental learning," in *IEEE International Conference on Computer Vision*, 2021.

[138] H. Kolb, E. Fernandez, and R. Nelson, "Webvision: the organization of the retina and visual system [internet]," 1995.

[139] M. Bear, B. Connors, and M. A. Paradiso, *Neuroscience: exploring the brain, enhanced edition: exploring the brain*. Jones & Bartlett Learning, 2020.

[140] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory," *Psychol Rev*, vol. 102, pp. 419–457, Jul 1995.

[141] D. Kumaran, D. Hassabis, and J. L. McClelland, "What Learning Systems do Intelligent Agents Need? Complementary Learning Systems Theory Updated," *Trends Cogn Sci*, vol. 20, pp. 512–534, Jul 2016.

[142] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proc Natl Acad Sci U S A*, vol. 114, pp. 3521–3526, Mar 2017.

[143] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 6470–6479, Curran Associates Inc., 2017.

[144] R. Kemker and C. Kanan, "Fearnet: Brain-inspired model for incremental learning," *arXiv preprint arXiv:1711.10563*, 2017.

[145] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[146] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, "How does the brain solve visual object recognition?," *Neuron*, 2012.

[147] A. Kohn, "Visual adaptation: physiology, mechanisms, and functional benefits," *J Neurophysiol*, 2007.

[148] V. V. Ramasesh, E. Dyer, and M. Raghu, "Anatomy of catastrophic forgetting: Hidden representations and task semantics," in *International Conference on Learning Representations Workshop*, 2021.

[149] J. New, L. Cosmides, and J. Tooby, "Category-specific attention for animals reflects ancestral priorities, not expertise," *Proceedings of the National Academy of Sciences*, vol. 104, no. 42, pp. 16598–16603, 2007.

[150] A. Borji, "Saliency prediction in the deep learning era: Successes, limitations, and future challenges," 2018.

[151] S. Treue and J. C. nez Trujillo, "Feature-based attention influences motion processing gain in macaque visual cortex," *Nature*, vol. 399, pp. 575–579, Jun 1999.

[152] J. C. Martinez-Trujillo and S. Treue, "Feature-based attention increases the selectivity of population responses in primate visual cortex," *Curr Biol*, vol. 14, pp. 744–751, May 2004.

[153] A. Linardos, M. Kümmerer, O. Press, and M. Bethge, "Deepgaze iie: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12919–12928, 2021.

[154] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1072–1080, 2015.

[155] R. Droste, J. Jiao, and J. A. Noble, "Unified image and video saliency modeling," in *European Conference on Computer Vision*, 2020.

[156] M. Boschini, L. Bonicelli, A. Porrello, G. Bellitto, M. Pennisi, S. Palazzo, C. Spampinato, and S. Calderara, "Transfer without forgetting," in *Proceedings of the European Conference on Computer Vision*, pp. 692–709, Springer, 2022.

[157] R. Aljundi, K. Kelchtermans, and T. Tuytelaars, "Task-free continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[158] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias, "Three types of incremental learning," *Nature Machine Intelligence*, 2022.

[159] Z. Mai, R. Li, H. Kim, and S. Sanner, "Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning," in *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, 2021.

[160] Y. Guo, B. Liu, and D. Zhao, "Online continual learning through mutual information maximization," in *International Conference on Machine Learning*, 2022.

[161] A. Zador, S. Escola, B. Richards, B. Ölveczky, Y. Bengio, K. Boahen, M. Botvinick, D. Chklovskii, A. Churchland, C. Clopath, J. DiCarlo, S. Ganguli, J. Hawkins, K. Koerding, A. Koulakov, Y. LeCun, T. Lillicrap, A. Marblestone, B. Olshausen, A. Pouget, C. Savin, T. Sejnowski, E. Simoncelli, S. Solla, D. Sussillo, A. S. Tolias, and D. Tsao, "Toward next-generation artificial intelligence: Catalyzing the neuroai revolution," *arXiv preprint*, 2022.

[162] S. Ebrahimi, S. Petryk, A. Gokul, W. Gan, J. E. Gonzalez, M. Rohrbach, and T. Darrell, "Remembering for the right reasons: Explanations reduce catastrophic forgetting," 2021.

[163] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, 2019.

[164] G. Saha and K. Roy, "Saliency guided experience packing for replay

in continual learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5273–5283, 2023.

[165] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[166] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," *Advances in neural information processing systems*, vol. 19, 2006.

[167] O. Le Meur, P. Le Callet, D. Barba, and D. Thoreau, "A coherent computational approach to model bottom-up visual attention," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 5, pp. 802–817, 2006.

[168] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *2009 IEEE 12th international conference on computer vision*, pp. 2106–2113, IEEE, 2009.

[169] X. Huang, C. Shen, X. Boix, and Q. Zhao, "Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 262–270, 2015.

[170] J. Pan, E. Sayrol, X. Giro-i Nieto, K. McGuinness, and N. E. O'Connor, "Shallow and deep convolutional networks for saliency prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 598–606, 2016.

[171] M. Kummerer, T. S. Wallis, L. A. Gatys, and M. Bethge, "Understanding low-and high-level contributions to fixation prediction," in *Proceedings of the IEEE international conference on computer vision*, pp. 4789–4798, 2017.

[172] W. Wang and J. Shen, "Deep visual attention prediction," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2368–2378, 2017.

[173] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto, "Salgan: Visual saliency prediction with generative adversarial networks," *arXiv preprint arXiv:1701.01081*, 2017.

[174] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an lstm-based saliency attentive model," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5142–5154, 2018.

[175] C. Bak, A. Kocak, E. Erdem, and A. Erdem, "Spatio-temporal saliency networks for dynamic saliency prediction," *IEEE Transactions on Multimedia*, vol. 20, no. 7, pp. 1688–1698, 2017.

[176] L. Bazzani, H. Larochelle, and L. Torresani, "Recurrent mixture density network for spatiotemporal visual attention," *arXiv preprint arXiv:1603.08199*, 2016.

[177] L. Jiang, M. Xu, and Z. Wang, "Predicting video saliency with object-to-motion cnn and two-layer convolutional lstm," *arXiv preprint arXiv:1709.06316*, 2017.

[178] W. Wang, J. Shen, F. Guo, M.-M. Cheng, and A. Borji, "Revisiting video saliency: A large-scale benchmark and a new model," in *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pp. 4894–4903, 2018.

[179] K. Min and J. J. Corso, "Tased-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2394–2403, 2019.

[180] G. Bellitto, F. Proietto Salanitri, S. Palazzo, F. Rundo, D. Giordano, and C. Spampinato, "Hierarchical domain-adapted feature learning for video saliency prediction," *International Journal of Computer Vision*, vol. 129, pp. 3216–3232, 2021.

[181] Z. Wang, Z. Liu, G. Li, Y. Wang, T. Zhang, L. Xu, and J. Wang, "Spatio-temporal self-attention network for video saliency prediction," *IEEE Transactions on Multimedia*, 2021.

[182] C. Ma, H. Sun, Y. Rao, J. Zhou, and J. Lu, "Video saliency forecasting transformer," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 10, pp. 6850–6862, 2022.

[183] X. Zhou, S. Wu, R. Shi, B. Zheng, S. Wang, H. Yin, J. Zhang, and C. Yan, "Transformer-based multi-scale feature integration network for video saliency prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[184] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[185] F. Hu, S. Palazzo, F. P. Salanitri, G. Bellitto, M. Moradi, C. Spampinato, and K. McGuinness, "Tinyhd: Efficient video saliency prediction with heterogeneous decoders using hierarchical maps distillation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.

[186] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proc. of the National Academy of Sciences*, 2017.

[187] N. Li, D. D. Cox, D. Zoccolan, and J. J. DiCarlo, "What response properties do individual neurons need to underlie position and clutter "invariant" object recognition?," *J Neurophysiol*, vol. 102, pp. 360–376, Jul 2009.

[188] T. Lesort, "Continual feature selection: Spurious features in continual learning," *arXiv preprint arXiv:2203.01012*, 2022.

[189] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[190] P. A. Lewis, G. Knoblich, and G. Poe, "How memory replay in sleep boosts creative problem-solving," *Trends in cognitive sciences*, vol. 22, no. 6, pp. 491–503, 2018.

[191] D. Kumaran, D. Hassabis, and J. L. McClelland, "What Learning Systems do Intelligent Agents Need? Complementary Learning

Systems Theory Updated," *Trends Cogn Sci*, vol. 20, pp. 512–534, Jul 2016.

[192] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory," *Psychol Rev*, vol. 102, pp. 419–457, Jul 1995.

[193] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy, *et al.*, "Dualprompt: Complementary prompting for rehearsal-free continual learning," in *Proceedings of the European Conference on Computer Vision*, pp. 631–648, Springer, 2022.

[194] D. Ji and M. A. Wilson, "Coordinated memory replay in the visual cortex and hippocampus during sleep," *Nat Neurosci*, vol. 10, pp. 100–107, Jan 2007.

[195] M. P. Walker and R. Stickgold, "Sleep-dependent learning and memory consolidation," *Neuron*, vol. 44, pp. 121–133, Sep 2004.

[196] D. Singh, K. Norman, and A. Schapiro, "A model of autonomous interactions between hippocampus and neocortex driving sleep-dependent memory consolidation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 119, Nov. 2022.

[197] M. Steriade, D. A. McCormick, and T. J. Sejnowski, "Thalamocortical oscillations in the sleeping and aroused brain," *Science*, vol. 262, pp. 679–685, Oct 1993.

[198] G. P. Krishnan, S. Chauvette, I. Shamie, S. Soltani, I. Timofeev, S. S. Cash, E. Halgren, and M. Bazhenov, "Cellular and neurochemical basis of sleep stages in the thalamocortical network," *Elife*, vol. 5, Nov 2016.

[199] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, "The "wake-sleep" algorithm for unsupervised neural networks," *Science*, vol. 268, pp. 1158–1161, May 1995.

[200] J. Bornschein and Y. Bengio, "Reweighted wake-sleep," *arXiv preprint arXiv:1406.2751*, 2014.

[201] T. Tadros, G. P. Krishnan, R. Ramyaa, and M. Bazhenov, "Sleep-like unsupervised replay reduces catastrophic forgetting in artificial neural networks," *Nat Commun*, vol. 13, p. 7742, Dec 2022.

[202] N. Deperrois, M. A. Petrovici, W. Senn, and J. Jordan, "Learning cortical representations through perturbed and adversarial dreaming," *Elife*, vol. 11, Apr 2022.

[203] M. J. Fosse, R. Fosse, J. A. Hobson, and R. J. Stickgold, "Dreaming and episodic memory: a functional dissociation?," *J Cogn Neurosci*, vol. 15, pp. 1–9, Jan 2003.

[204] S. Llewellyn, "Dream to Predict? REM Dreaming as Prospective Coding," *Front Psychol*, vol. 6, p. 1961, 2015.

[205] S. Schwartz, "Are life episodes replayed during dreaming?," *Trends Cogn Sci*, vol. 7, pp. 325–327, Aug 2003.

[206] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly,

J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021.

[207] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-inspired artificial intelligence," *Neuron*, vol. 95, no. 2, pp. 245–258, 2017.

[208] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, and T. Masquelier, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, pp. 47–63, 2019.

[209] G. Tononi and C. Cirelli, "Sleep and the price of plasticity: from synaptic and cellular homeostasis to memory consolidation and integration," *Neuron*, vol. 81, no. 1, pp. 12–34, 2014.

[210] L. Marshall and J. Born, "The contribution of sleep to hippocampus-dependent memory consolidation," *Trends in cognitive sciences*, vol. 11, no. 10, pp. 442–450, 2007.