Wake-Sleep Consolidated Learning

Amelia Sorrenti, Giovanni Bellitto, *Member, IEEE*, Federica Proietto Salanitri, *Member, IEEE*, Matteo Pennisi, Simone Palazzo, and Concetto Spampinato, *Member, IEEE*

***Abstract*— We propose wake-sleep consolidated learning (WSCL), a learning strategy leveraging complementary learning system (CLS) theory and the wake-sleep phases of the human brain to improve the performance of deep neural networks (DNNs) for visual classification tasks in continual learning (CL) settings. Our method learns continually via the synchronization between distinct wake and sleep phases. During the wake phase, the model is exposed to sensory input and adapts its representations, ensuring stability through a dynamic parameter freezing mechanism and storing episodic memories in a short-term temporary memory (similar to what happens in the hippocampus). During the sleep phase, the training process is split into nonrapid eye movement (NREM) and rapid eye movement (REM) stages. In the NREM stage, the model's synaptic weights are consolidated using replayed samples from the short-term and long-term memory and the synaptic plasticity mechanism is activated, strengthening important connections and weakening unimportant ones. In the REM stage, the model is exposed to previously-unseen realistic visual sensory experience, and the dreaming process is activated, which enables the model to explore the potential feature space, thus preparing synapses for future knowledge. We evaluate the effectiveness of our approach on four benchmark datasets: CIFAR-10, CIFAR-100, Tiny-ImageNet, and FG-ImageNet. In all cases, our method outperforms the baselines and prior work, yielding a significant performance gain on continual visual classification tasks. Furthermore, we demonstrate the usefulness of all processing stages and the importance of dreaming to enable positive forward transfer (FWT). The code is available at: https://github.com/perceivelab/wscl.**

***Index Terms*— Complementary learning systems (CLSs), continual learning (CL), off-line brain states.**

I. Introduction

HUMANS have a remarkable ability to continuously learn and retain past experiences while quickly adapting to new tasks and problems. On the contrary, machine learning has shown limitations when dealing with nonstationary data streams. This can be attributed to the inherent structure and optimization approaches of artificial neural networks, which differ significantly from how humans learn and build neural connectivity over a lifetime. The complementary learning systems (CLSs) theory [1], [2] 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 (CL) methods [3], [4] which translate CLS concepts into computational frameworks. DualNet [3] 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 [4] 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 artificial 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 [5], [6], [7]. During sleep, neurons are spontaneously active without external input and generate complex patterns of synchronized activity across brain regions [8], [9]. This strong neural activity is believed to be due to the brain replaying and consolidating memories while reorganizing synaptic connections [10].

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' CL 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) is used to emulate 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; and 2) a sleep phase, consisting of two alternating stages: a) nonrapid 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; and b) rapid eye movement (REM), where dreaming simulates new experience, preparing the brain for future events. The hypothesis is that

Received 3 August 2023; revised 10 May 2024 and 4 September 2024; accepted 6 September 2024. The work of Giovanni Bellitto, Federica Proietto Salanitri, and Concetto Spampinato was supported by PNRR MUR Project PE0000013-FAIR. *(Corresponding author: Amelia Sorrenti.)*

The authors are with the PeRCeiVe Lab Research Group, University of Catania, 95123 Catania, Italy (e-mail: amelia.sorrenti@phd.unict.it).

Digital Object Identifier 10.1109/TNNLS.2024.3458440

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, the WSCL approach is the first CL method yielding positive forward transfer (FWT), demonstrating its ability to prepare synapses for 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.

II. Related Work

CL [11], [12] is a branch of machine learning whose objective is to bridge the gap in incremental learning between humans and neural networks. McCloskey and Cohen [13] highlight that the latter undergo catastrophic forgetting of previously acquired knowledge in the presence of input distribution shifts. To mitigate this problem, various strategies have been proposed, encompassing the integration of appropriate regularization terms [14], [15], tailored architectural configurations [16], [17], and the utilization of rehearsal mechanisms involving a limited set of previously encountered data points [18], [19], [20]. Notably, rehearsal-based approaches emerge as the most promising avenue for combating catastrophic forgetting in CL scenarios, especially in dynamic real-world applications. Unlike static models prone to overwriting prior knowledge, rehearsal-based techniques capitalize on past experiences by storing select samples in a small buffer, which the model continues to train on even when presented with new tasks.

While current solutions help reduce forgetting, real-world application proves difficult, as typical CL evaluations are carried out on oversimplistic benchmarks [21], [22]. Most approaches tackling this challenging scenario combine a replay strategy [18], [23], [24], [25] to regularization on logits sampled throughout the optimization trajectory [20]. Some works focus on memory management: GSS [26] introduces a specific optimization of the basic rehearsal formula meant to store maximally informative samples; HAL [27] individuates synthetic replay data points that are maximally affected by forgetting. CaSpeR [25] adopts a rehearsal-based strategy enforcing latent space regularization on buffer samples through geometric constraints. Other works propose tailored classification schemes: CoPE [28] uses class prototypes to ensure a gradual evolution of the shared latent space; ER-ACE [29] makes the cross-entropy loss asymmetric to minimize imbalance between current and past tasks. Recent works introduce a surrogate optimization objective: CR [30] employs a supervised contrastive learning objective and OCM [31] 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 (CLSs and wake-sleep) and exploit 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) [14] and synaptic intelligence [15] 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 [32] adopts an auxiliary network (in line with CLS theory) to detect catastrophic forgetting and trigger knowledge-preserving regularization. Co2L [33] learns stable representations through contrastive learning and self-supervised distillation.

Existing approaches similarly inspired by CLS theory are DualNet [3], DualPrompt [4], and CLS-ER [34]. DualNet employs two networks that loosely emulate slow and fast learning in humans. DualPrompt [4] also takes a cognitive approach, using learnable prompts to be paired to a pretrained transformer backbone. CLS-ER [34] implements semantic memory using two separate neural networks to model short-term and long-term memory dynamics. While these approaches yield good results, they ignore off-line states, that appear fundamental in human learning. Alternating between wake and sleep phases has already been shown to have the potential for learning improved and robust semantic representations [35], [36]. Sleep replay consolidation [37] employs sleep-based training using local unsupervised Hebbian plasticity rules for mitigating catastrophic forgetting of ANN. The recent SIESTA [38] introduces alternating waking and sleeping and it primarily focuses on online learning with intermittent consolidation phases.

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 [10], 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 [10] and [37] does not seem sufficient to build robust representations during sleep [7]: our work thus explores more sophisticated restructuring of neural connections in the neocortex guided by the hippocampus.

Finally, WSCL utilizes a selecting freezing strategy of the model's parameters. This strategy aims exclusively to enhance model efficiency, mirroring human fast adaptation, and operates during training without task-specific parameter selection during inference. This strategy is fundamentally different from architectural approaches such as the ones in [16], [17], [39], [40], and [41] that, instead, learn task-specific parameters which are then selected, at inference time, based on task identifiers [17], mask entropy [40], or scaling parameters [41]: WSCL maintains a unified network for all tasks, eliminating the necessity for task-specific masks or parameters during inference.

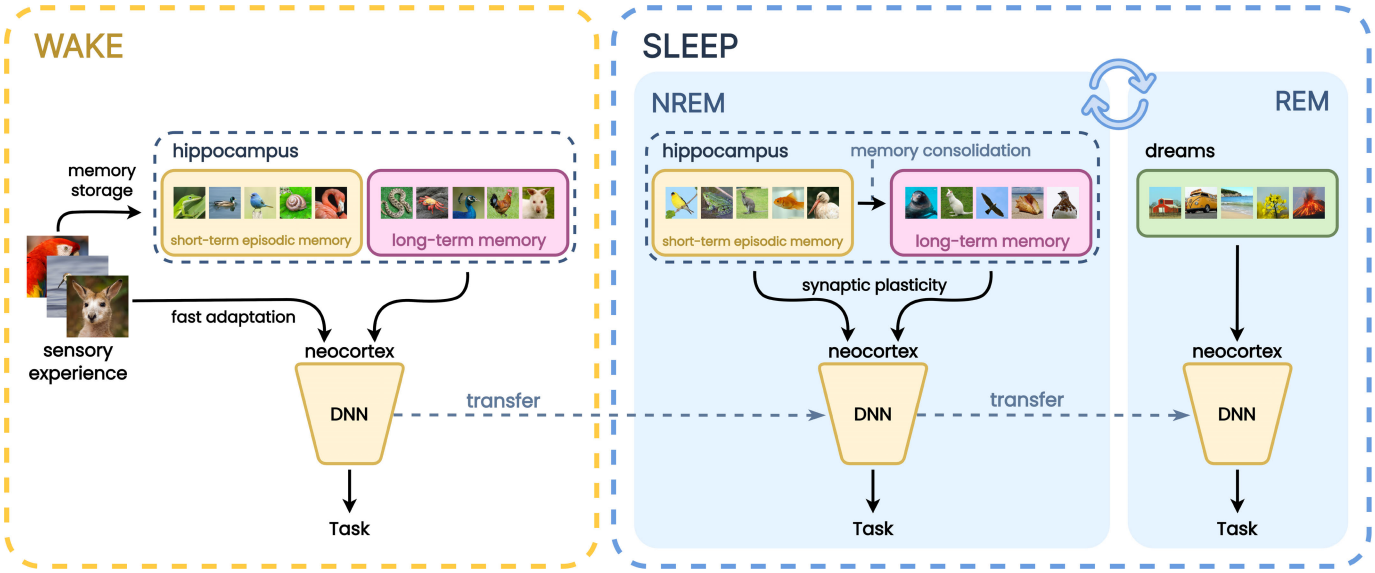


Fig. 1. WSCL: in the wake stage, the model (which emulates the neocortex) fast adapts to the new sensory experience, storing episodic memories (as in the hippocampus) in the short-term memory to be replayed during sleep. The sleep phase foresees two alternating processes: 1) the NREM stage, where the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

III. Method

An overview of the WSCL approach is presented in Fig. 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 a 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 quick learning general and transferable representation by combining both current and past experience as well as in identifying which part of the network has to be trained and which not. In the sleep phase, the model consolidates newly acquired knowledge by revisiting the hippocampus's short-term memory containing the task data, merging it into existing knowledge by updating synaptic connections, moving it into 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 with dreaming.

Following the established literature, we pose CL as a supervised classification problem on a non-i.i.d. stream of data, with the assumption that task boundaries, and 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 $\mathcal{D}_i$ is different from those from $\mathcal{D}_j$. Given a classifier $f : \mathcal{X} \to \mathcal{Y}$, 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 τ_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

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

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

$$\arg\min_{\boldsymbol{\theta}_T} \sum_{i=1}^{T} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i} \big[\mathcal{L}(f(\mathbf{x}; \boldsymbol{\theta}_T), y) \big] \tag{2}$$

where $\mathcal{L}$ is a generic classification loss (e.g., cross-entropy), which a CL model attempts to optimize while accounting for model plasticity (the capability to learn current task data) and stability (the capability to retain the knowledge of previous tasks) [13].

A. 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 CL setting, we model short-term memorization into $\mathbf{M}_s$ as a sampling of task data $\mathcal{D}_i$

$$\mathbf{M}_s = \left\{\left(\mathbf{x}_j, y_j\right) \sim \mathcal{D}_i\right\}_{j=1}^{N_s} \tag{3}$$

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

Fast model adaptation. In accordance with CLS theory [1], [2], we propose a method for fast model adaptation that employs parameter freezing during the wake stage to maximize stability and plasticity. This strategy operates only during the model's training, as it aims at adapting quickly the model to the current data distribution, while it does not activate at inference time when the model's knowledge is already consolidated. Specifically, fast model adaption works by training the model for a limited number of iterations under varying parameter freezing settings, providing an opportunity for the model to rapidly learn new information in the wake stage while retaining the previous knowledge; in-depth consolidation of task information is 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

$$P(\mathbf{x}, y, \boldsymbol{\theta}_i, \mathbf{m}_i) = P(y \mid \mathbf{x}, f(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i)) P(\boldsymbol{\theta}_i, \mathbf{m}_i) P(\mathbf{x}) \tag{4}$$

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 (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 distribution can be expressed as

$$P(\boldsymbol{\theta}_i, \mathbf{m}_i) = P(\boldsymbol{\theta}_i \mid \mathbf{m}_i) P(\mathbf{m}_i) \tag{5}$$

where

$$P(\boldsymbol{\theta}_i \mid \mathbf{m}_i) = \prod_j \mathcal{N}\left(\theta_{i,j}; \theta_{i-1,j}, \sigma_i^2\right)^{1-m_{i,j}}. \tag{6}$$

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).

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

However, modeling $P(\mathbf{m}_i)$ by operating on each parameter $\theta_{i,j}$ is practically infeasible, we thus employ some simplifying assumptions 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 $\boldsymbol{\theta} = [\boldsymbol{\theta}_{|1}, \ldots, \boldsymbol{\theta}_{|L}]$, 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:

$$\mathbf{m}_i(l) = \left[\mathbf{1}_{|1}, \ldots, \mathbf{1}_{|l}, \mathbf{0}_{|l+1}, \ldots, \mathbf{0}_{|L}\right] \vee \mathbf{m}_{i-1} \tag{7}$$

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:

$$\arg\max_{\mathbf{m}_i, \boldsymbol{\theta}_i} P(y \mid \mathbf{x}, f(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i)) P(\boldsymbol{\theta}_i \mid \mathbf{m}_i) P(\mathbf{m}_i) P(\mathbf{x}) \tag{8}$$

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 $\mathcal{L}$

$$\begin{aligned}\mathcal{L}_{\text{fma}} = {} & \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}\left[\mathcal{L}(y, f(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i))\right] \\ & + \alpha \mathbb{E}_{(\mathbf{x},y)\sim\mathbf{M}_{i-1}}\left[\mathcal{L}(y, f(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i))\right]\end{aligned} \tag{9}$$

where $\mathbf{m}_i$ varies as described above, and α 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 CL method, enhancing it with the proposed training strategy.

B. Sleep Phase

During sleep, the brain cycles multiple times through two phases, known as REM and 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 [5], [42]. REM sleep is thought to play a role in creativity and problem-solving [43], [44], 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 stabilize 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 CL task. This approach allows the model to learn task-agnostic features which can be interpreted as prior knowledge supporting task-specific learning and FWT. 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 with 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 (4), but optimize for $\boldsymbol{\theta}_i$ alone, while leaving $\mathbf{m}_i$ constant. The objective thus becomes

$$\arg\max_{\boldsymbol{\theta}_i} P(y \mid \mathbf{x}, f(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i)) P(\boldsymbol{\theta}_i \mid \mathbf{m}_i) P(\mathbf{x}) \tag{10}$$

where the prior on parameters $P(\boldsymbol{\theta}_i|\mathbf{m}_i)$ is essentially the same as in (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 (9)

$$\begin{aligned}\mathcal{L}_{\text{NREM}} = \mathbb{E}_{(\mathbf{x},y)\sim\mathbf{M}_s}\big[\mathcal{L}(y, f(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i))\big] \\ + \alpha \mathbb{E}_{(\mathbf{x},y)\sim\mathbf{M}_{i-1}}\big[\mathcal{L}(y, f(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i))\big]\end{aligned} \tag{11}$$

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 [45] 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 [10].

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 CL classes), to prevent any overlap between auxiliary and CL classes. Given this premise, the proposed optimization objective becomes

$$\arg\max_{\boldsymbol{\theta}_i} P(y \mid \mathbf{x}, f(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i)) P(\boldsymbol{\theta}_i \mid \mathbf{m}_i) P(\mathbf{x}) \tag{12}$$

where $(\mathbf{x}, y) \sim \mathcal{D}_{\text{dream}}$, while the other terms are the same as in (10). This objective is then mapped to a training loss function defined as

$$\mathcal{L}_{\text{REM}} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{\text{dream}}}\big[\mathcal{L}(y, f(\mathbf{x}, \boldsymbol{\theta}_i, \mathbf{m}_i))\big]. \tag{13}$$

During the REM stage, training with two distinct class label sets, $\mathcal{Y}$ from the CL problem and $\mathcal{Y}_{\text{dream}}$ from the dreaming dataset has been addressed following the procedure reported in [46].

IV. Experimental Evaluation

A. Benchmarks

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

1) *Split CIFAR-10 [15]:* A widely used image classification dataset obtained by splitting CIFAR-10 images into five 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.
2) *Split CIFAR-100 [15]:* Which is a variant of the CIFAR-100 dataset where the original 100 classes are divided into ten disjoint subsets to test CL performance. Each subset, or split, contains a portion of the original classes. Its counterpart used for the REM stage (referred to as ImageNet$^{\text{aux}}$) consists of a subset of 100 ImageNet classes, selected after removing those with semantic relations to CIFAR-100.
3) *Split 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 (referred as to ImageNet$^{\text{NO}}$) consists of additional 100 classes taken from ImageNet, after removing all synsets derived from "organism" ($^{\text{NO}}$ stands for "nonorganism").
4) *Tiny-ImageNet* [47]: 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 tinyimagenet (organized as five tasks of 20 classes) and the remaining 100 classes as the dreaming dataset (referred to in the article as to Tiny-ImageNet^{D}).

B. Training Procedure

Our approach employs a ResNet-18 backbone for feature extraction and classification. ResNet-18 includes, at a high level, four layers with two blocks each, for a total of eight blocks.[3] With reference to the definition of model f in Section III-A, we map each layer l_i to each ResNet-18 block.

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 a 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 α hyperparameter in (9) is set to 1, and the N_s dimension of the short-term buffer to 5000.

It is important to mention that, in our implementation, the optimization of (9) (fast model adaptation loss $\mathcal{L}_{\text{fma}}$) and (11) (NREM loss $\mathcal{L}_{\text{NREM}}$) on long-term memory $\mathbf{M}_i$ is carried

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

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

TABLE I
CLASS-INCREMENTAL FAA OF REHEARSAL-BASED METHODS, WITH AND WITHOUT WSCL, FOR DIFFERENT BUFFER SIZES

Target dataset	**CIFAR-10**		**CIFAR-100**		**Tiny-ImageNet$_{1/2}$**		**FG-ImageNet**	
Dreaming dataset	*CIFAR-100*		*ImageNet^{aux}*		*Tiny-ImageNet^{D}*		*ImageNet^{NO}*	
Buffer size	**200**	**500**	**200**	**500**	**200**	**500**	**200**	**1000**
GDumb [48]	33.81 ± 1.52	46.01 ± 3.26	6.71 ± 1.55	10.95 ± 0.20	5.74 ± 0.45	9.85 ± 0.50	4.54 ± 0.23	9.79 ± 1.18
↪**WSCL**	**66.85** ± 1.60	**72.35** ± 0.72	**14.79** ± 0.28	**23.68** ± 0.72	**17.79** ± 2.34	**26.84** ± 0.84	**7.70** ± 0.14	**16.43** ± 0.10
ER [29]	48.76 ± 0.57	59.75 ± 2.51	14.31 ± 0.47	20.71 ± 0.95	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	**16.91** ± 1.26	**24.25** ± 0.44	**18.81** ± 0.48	**23.63** ± 0.85	**6.01** ± 0.64	**15.26** ± 3.59
DER++ [20]	57.35 ± 5.47	69.06 ± 1.24	14.93 ± 2.23	23.26 ± 3.19	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	**24.00** ± 0.94	**31.96** ± 1.19	**23.70** ± 0.91	**31.81** ± 0.70	**6.48** ± 1.22	**11.70** ± 0.14
ER-ACE [24]	59.98 ± 2.65	67.17 ± 1.54	25.85 ± 1.93	32.85 ± 4.02	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	**34.44** ± 0.40	**39.78** ± 0.36	**35.68** ± 1.18	**41.25** ± 1.75	**12.51** ± 0.86	**20.51** ± 0.56
DualNet [3]	31.31 ± 2.05	43.20 ± 2.81	9.49 ± 0.28	11.04 ± 4.39	16.45 ± 0.39	18.98 ± 0.71	9.78 ± 1.24	16.54 ± 0.85
CoPE [28]	21.20 ± 0.28	23.64 ± 1.56	13.71 ± 0.43	21.03 ± 0.49	16.50 ± 0.62	20.50 ± 0.47	6.23 ± 0.61	12.57 ± 3.69
CLS-ER [34]	34.97 ± 4.83	45.17 ± 4.20	23.85 ± 1.16	30.22 ± 0.76	15.38 ± 0.43	18.19 ± 0.85	9.06 ± 1.32	15.15 ± 1.48

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 alternate batches. We perform 10 epochs of training, with the same optimizer settings and hyperparameters as above.

All the reported results are computed in the class-incremental setting and reported as mean and standard deviation computed over five runs.

C. Results

We first evaluate how WSCL contributes to the classification accuracy of state-of-the-art models. To accomplish this, we select recent rehearsal-based methods, namely, DER++ [20], ER-ACE [29] and ER [24], and compare their performance when the WSCL training strategy is employed, by plugging them in as the $\mathcal{L}$ loss term in (9), (11), and (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 setting. We further provide a lower bound, consisting of training without any countermeasure to forgetting (fine-tuning), and an upper bound given by training all tasks jointly (Joint). Results in Table I show that, on all four 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 I also reports the comparison with.

1) DualNet [3], which leverages CLS theory and the same backbone, i.e., ResNet-18.
2) CoPE [28] that integrates contrastive learning—another technique inspired by cognitive neuroscience [10]—for better feature transferability to later tasks.[4]
3) CLS-ER [34], another method inspired by CLS theory that implements a semantic memory with two separate DNNs to model short-term and long-term memory dynamics.

We do not include DualPrompt [4] as it uses a large pretrained ViT [49] 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), CoPE, and CLS-ER, demonstrating how mimicking off-line brain states improves performance even in a purely discriminative supervised learning regime. We also measure the FWT of WSCL, a desirable property in CL that indicates how much a model leverages previous knowledge to learn a new task [45]. FWT is estimated as the average difference between the accuracy of a task when learning it in a CL setting and when learning it from random initialization (details in [45]). Table II shows how WSCL tends to enhance FWT, turning it from negative to positive values. This is highly remarkable as the majority of existing CL methods show a negative FWT.

Furthermore, it is equally important to measure forgetting (the lower, the better) to assess how well an approach tackles non-iid data. Cross-checking results in Table III with those available in Table I highlights how WSCL effectively reduces forgetting, while enhancing FWT skills and accuracy performance in a way sensibly higher than the baselines.

Previous results have primarily been derived from experiments conducted on common CL benchmarks, where dreaming datasets typically exhibit a semantic distribution shift, meaning their image classes do not overlap with those of the target task. In order to further validate the efficacy of the wake-sleep CL (WSCL) strategy, we extend our evaluation to scenarios involving domain distribution shifts. Specifically, we assess

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

TABLE II
CLASS-INCREMENTAL FWT OF REHEARSAL-BASED METHODS, WITH AND WITHOUT WSCL, FOR DIFFERENT BUFFER SIZES

Target dataset	**CIFAR-10**		**CIFAR-100**		**Tiny-ImageNet$_{1/2}$**		**FG-ImageNet**	
Dreaming dataset	*CIFAR-100*		*ImageNet^{aux}*		*Tiny-ImageNet^{D}*		*ImageNet^{NO}*	
Buffer size	**200**	**500**	**200**	**500**	**200**	**500**	**200**	**1000**
ER [29]	-7.36 ± 5.75	-12.20 ± 0.74	-0.80 ± 0.48	-0.92 ± 0.52	-1.00 ± 0.33	-1.32 ± 0.08	-1.05 ± 0.06	-1.02 ± 0.09
↪**WSCL**	1.68 ± 2.16	6.03 ± 2.58	7.07 ± 1.37	6.49 ± 2.01	12.41 ± 0.98	12.60 ± 0.43	3.82 ± 0.64	3.17 ± 0.73
DER++ [20]	-12.29 ± 0.18	-6.23 ± 5.63	-0.87 ± 0.51	-0.72 ± 0.43	-0.84 ± 0.49	-1.06 ± 0.47	-0.08 ± 0.00	-1.05 ± 0.42
↪**WSCL**	1.06 ± 6.05	2.83 ± 5.27	8.83 ± 0.30	7.52 ± 0.84	12.16 ± 1.25	12.24 ± 1.52	1.78 ± 0.43	2.31 ± 0.07
ER-ACE [24]	-8.58 ± 5.03	-8.97 ± 2.21	-1.01 ± 0.61	-1.02 ± 0.14	-0.73 ± 0.51	-0.94 ± 0.47	-1.04 ± 0.02	-1.17 ± 0.14
↪**WSCL**	0.48 ± 5.53	-1.87 ± 3.12	6.25 ± 0.81	6.06 ± 0.43	8.60 ± 0.96	9.06 ± 1.22	1.83 ± 0.69	1.19 ± 0.67

TABLE III
CLASS-INCREMENTAL FORGETTING OF REHEARSAL-BASED METHODS, WITH AND WITHOUT WSCL, FOR DIFFERENT BUFFER SIZES

Target dataset	**CIFAR-10**		**CIFAR-100**		**Tiny-ImageNet$_{1/2}$**		**FG-ImageNet**	
Dreaming dataset	*CIFAR-100*		*ImageNet^{aux}*		*Tiny-ImageNet^{D}*		*ImageNet^{NO}*	
Buffer size	**200**	**500**	**200**	**500**	**200**	**500**	**200**	**1000**
ER [29]	56.66 ± 2.64	43.21 ± 3.41	73.13 ± 1.04	65.49 ± 1.78	62.63 ± 3.43	58.16 ± 1.13	74.04 ± 2.09	73.45 ± 2.06
↪**WSCL**	50.23 ± 4.94	36.04 ± 2.52	68.66 ± 1.04	60.80 ± 1.60	56.71 ± 1.68	50.63 ± 1.37	76.79 ± 0.67	63.93 ± 0.91
DER++ [20]	31.23 ± 2.07	22.63 ± 1.68	67.76 ± 2.87	54.76 ± 5.15	62.15 ± 1.27	50.81 ± 2.56	67.10 ± 2.83	63.63 ± 4.12
↪**WSCL**	35.53 ± 3.28	23.52 ± 2.29	54.63 ± 1.42	46.33 ± 0.81	51.30 ± 2.26	43.91 ± 0.75	59.84 ± 1.59	52.39 ± 2.01
ER-ACE [24]	16.55 ± 2.03	15.21 ± 2.05	38.37 ± 0.86	30.77 ± 6.28	34.41 ± 1.35	28.15 ± 1.96	32.61 ± 3.56	36.44 ± 2.46
↪**WSCL**	11.78 ± 1.61	10.69 ± 2.02	28.20 ± 1.05	25.91 ± 0.89	28.23 ± 1.52	23.29 ± 4.47	27.24 ± 0.55	33.53 ± 0.76

TABLE IV
WSCL PERFORMANCE ON THE CORE50 DATASET WITH MULTIPLE DREAMING DATASETS, IN TERMS OF ACCURACY (FAA): WSCL LEADS TO SIGNIFICANT PERFORMANCE GAINS, DEMONSTRATING ITS EFFICACY

Target dataset	**CORe50**							
Buffer size	**200**				**500**			
ER [29]	19.97 ± 0.02				19.93 ± 0.10			
DER++ [20]	19.88 ± 0.05				19.94 ± 0.06			
ER-ACE [24]	19.90 ± 0.03				19.96 ± 0.02			
Dreaming dataset	*CIFAR-10*		*CIFAR-100*		*Tiny-ImageNet^{D}*		*ImageNet^{NO}*	
Buffer size	**200**	**500**	**200**	**500**	**200**	**500**	**200**	**500**
ER [29] ↪**WSCL**	**36.40** ± 1.91	**44.03** ± 0.79	**34.40** ± 3.33	**51.71** ± 2.71	**37.30** ± 2.44	**56.07** ± 4.00	**43.89** ± 2.93	**60.94** ± 2.15
DER++ [20] ↪**WSCL**	**47.43** ± 2.41	**54.31** ± 4.28	**41.83** ± 2.57	**58.83** ± 1.25	**53.13** ± 4.99	**63.85** ± 2.34	**60.43** ± 3.86	**71.20** ± 5.01
ER-ACE [24] ↪**WSCL**	**59.46** ± 3.93	**68.48** ± 0.67	**61.62** ± 1.68	**71.12** ± 0.76	**60.53** ± 2.19	**71.02** ± 3.30	**66.85** ± 1.53	**76.94** ± 2.57

the performance of the WSCL strategy, when combined with ER, DER++, and ER-ACE, on the challenging CORe50 dataset [50] with multiple dreaming datasets. As shown in Table IV, our approach achieves exceptional performance gains, with improvements of up to 50 percent points, across all dreaming datasets. These results not only underscore the effectiveness of the WSCL strategy but also highlight its applicability to extremely complex datasets where conventional approaches often fall short.

We further expand performance analysis by grounding WSCL to other prominent CL methods.[5] For this evaluation, we employ of model that yields the highest results, i.e., ER-ACE combined to WSCL (as shown in Table I). As shown in Table V, ER-ACE w/ WSCL (indicated as "ours") significantly outperforms all existing methods. Notably, when excluding the buffer for training ER-ACE w/ WSCL (which means using the model without NREM stage, indicated in Table V as Wake + REM), it achieves a substantial performance improvement, from approximately 12% (on Tiny-ImageNet$_{1/2}$) to about 23% on CIFAR-10, over existing buffer-free methods, namely, LwF [51], SI [15], and oEWC [16].

[5]Results obtained using the original code released along with the relative papers.

D. Model Analysis

The model analysis primarily utilizes ER-ACE, identified as the top-performing method (refer to Table I), as the baseline. We evaluate its performance on both the Tiny-ImageNet$_{1/2}$ and CORe50 datasets, examining scenarios involving semantic and domain shifts. Initially, we conduct ablations on the processing phases of WSCL. Results in Table VI demonstrate that NREM and REM sleep states contribute equally to the final model performance across both benchmarks.

TABLE V
COMPARISON WITH SOTA METHODS, IN TERMS OF CLASS-INCREMENTAL FAA, FOR DIFFERENT BUFFER SIZES

Method	**CIFAR-10**		**CIFAR-100**		**Tiny-ImageNet$_{1/2}$**		**FG-ImageNet**	
Joint	85.15 ± 1.99		61.83 ± 0.47		50.81 ± 1.65		43.39 ± 1.76	
Fine-tune	19.47 ± 0.10		9.11 ± 0.11		13.84 ± 0.55		3.88 ± 0.33	
Buffer-free methods								
LwF [51]	19.33 ± 0.16		8.44 ± 0.39		13.87 ± 1.11		3.83 ± 0.11	
SI [15]	19.27 ± 0.23		7.86 ± 1.08		13.12 ± 1.63		3.75 ± 0.35	
oEWC [16]	18.96 ± 0.24		7.11 ± 0.40		13.87 ± 0.53		3.48 ± 0.18	
Ours (Wake+REM)	**41.58** ± 3.94		**18.17** ± 1.04		**25.68** ± 0.44		**6.27** ± 0.89	
Buffer-based methods								
Buffer size	**200**	**500**	**200**	**500**	**200**	**500**	**200**	**1000**
ER [29]	48.76 ± 0.57	59.75 ± 2.51	14.31 ± 0.47	20.71 ± 0.95	16.25 ± 0.85	21.07 ± 1.43	4.23 ± 0.15	5.05 ± 0.51
DER++ [20]	57.35 ± 5.47	69.06 ± 1.24	14.33 ± 1.97	23.26 ± 3.32	16.62 ± 1.76	23.40 ± 1.66	5.95 ± 0.49	8.59 ± 1.11
ER-ACE [24]	59.98 ± 2.65	67.17 ± 1.54	25.86 ± 1.94	32.85 ± 4.02	27.81 ± 1.24	32.10 ± 2.21	9.42 ± 0.78	11.58 ± 3.59
A-GEM [52]	19.45 ± 0.25	20.21 ± 0.38	8.34 ± 0.97	8.02 ± 1.25	13.75 ± 0.37	13.56 ± 0.39	4.00 ± 0.20	4.15 ± 0.06
BiC [53]	55.03 ± 1.93	66.24 ± 1.65	21.80 ± 3.81	29.52 ± 2.06	16.26 ± 0.87	12.88 ± 5.50	8.10 ± 2.75	7.03 ± 4.71
FDR [54]	38.72 ± 8.93	31.91 ± 5.08	13.41 ± 1.02	19.34 ± 2.29	17.67 ± 1.04	23.17 ± 1.69	4.44 ± 0.77	3.91 ± 0.22
GEM [45]	21.93 ± 2.04	20.80 ± 0.23	10.40 ± 2.19	14.39 ± 4.11	14.57 ± 0.57	15.20 ± 1.28	4.36 ± 0.11	4.29 ± 0.28
GDumb [48]	33.81 ± 1.52	46.01 ± 3.26	6.71 ± 1.55	10.95 ± 0.20	5.74 ± 0.45	9.85 ± 0.50	4.54 ± 0.23	9.79 ± 1.18
GSS [26]	41.36 ± 6.46	48.83 ± 4.41	11.11 ± 0.63	12.78 ± 0.18	15.92 ± 0.88	18.15 ± 0.61	4.05 ± 0.42	4.46 ± 1.20
iCaRL [19]	64.52 ± 1.18	60.94 ± 1.34	14.22 ± 0.22	16.01 ± 0.52	20.40 ± 0.36	22.68 ± 0.30	10.40 ± 0.20	11.17 ± 0.79
LUCIR [55]	53.48 ± 7.62	63.01 ± 3.40	24.06 ± 1.81	32.54 ± 1.16	22.65 ± 1.18	32.15 ± 0.88	6.08 ± 0.32	13.19 ± 0.32
RPC [56]	49.37 ± 1.47	55.19 ± 2.73	14.38 ± 1.36	21.01 ± 0.95	16.58 ± 0.52	20.95 ± 0.59	4.13 ± 0.16	5.83 ± 0.30
Ours	**71.15** ± 2.15	**74.18** ± 1.28	**34.44** ± 0.40	**39.78** ± 0.36	**35.68** ± 1.18	**41.25** ± 1.75	**12.51** ± 0.86	**20.51** ± 0.56

TABLE VI
ABLATION ON THE WSCL PROCESSING STAGES: RESULTS REFER TO ER-ACE ON TINY-IMAGENET$_{1/2}$ AND ON CORE50 DATASETS WITH BUFFER SIZE OF 200

Target dataset	**Tiny-ImageNet$_{1/2}$**		**CORe50**	
Dreaming dataset	*Tiny-Imagenet^D*			
Method	**FAA**	**FWT**	**FAA**	**FWT**
Only Wake	4.70	-0.93	15.00	-7.49
Wake + REM	25.68	11.89	23.82	-11.80
Wake + NREM	27.61	-0.67	54.15	-12.61
Wake + REM + NREM	35.68	8.60	60.53	-5.62

TABLE VII
COMPARISON OF CLASSIFICATION PERFORMANCE IN TERMS OF ACCURACY (FAA) BETWEEN WSCL AND CASPER. RESULTS ARE COMPUTED ONLY IN COMBINATION WITH DER++ AND ER-ACE AS THE OTHER MODELS IN [25] MANIPULATE FUTURE LOGITS HINDERING WSCL APPLICATION. DREAMING DATASETS ARE CIFAR-100 AND IMAGENET^{AUX}, RESPECTIVELY, FOR CIFAR-10 AND CIFAR-100 BENCHMARKS. BUFFER SIZE IS 500

Dataset	**CIFAR-10**	**CIFAR-100**
DER++ [20]	67.38	28.01
↪CaSpeR [25]	69.11	32.16
↪**WSCL**	**72.18**	**35.00**
ER-ACE [24]	66.13	34.99
↪CaSpeR [25]	69.58	36.70
↪**WSCL**	**73.56**	**39.33**

Notably, on the Tiny-ImageNet$_{1/2}$ dataset, the REM phase exhibits positive FWT, aligning with cognitive neuroscience findings indicating REM's role in priming brain synapses for future experiences [43], [44]. This pattern of FWT is observed across the majority of the tested datasets (see Table II), except for CORe50. This deviation can be attributed to the nature of the CORe50 benchmark, which inherently restricts feature reuse across tasks due to its strong background bias.

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. 2 indicates that WSCL still outperforms the baseline when dreaming images are affected by noise up to 30% or scaled down by 6×, suggesting that the role of REM stage in consolidating knowledge is mostly independent of the visual details of the dreamed samples, which merely serve to learn additional reusable features.

We further investigate the impact of the size of the dreaming dataset on the results. Fig. 3 illustrates how the dreaming stage allows for enhanced performance even when the additional dreaming dataset is reduced by approximately 70%.

Dreaming in WSCL serves as an implicit regularizer for the learned latent space within the model, maintaining consistent partitioning across classes, especially on low-dimensional buffers. In this study, we contrast our WSCL approach with a regularization method, namely, CaSpeR [25], which explicitly encourages partitioning behavior in the learned space by imposing constraints on its Laplacian spectrum. To ensure a fair comparison, we replicated the experimental setup detailed in [25], conducting tests over the same number of epochs (20) and batch size of 64 on the CIFAR-10 and CIFAR-100 datasets with buffer 500 (as these represent the intersection between [25] evaluation and ours). The results, presented in Table VII, highlight how WSCL outperforms CaSpeR as a regularizer.

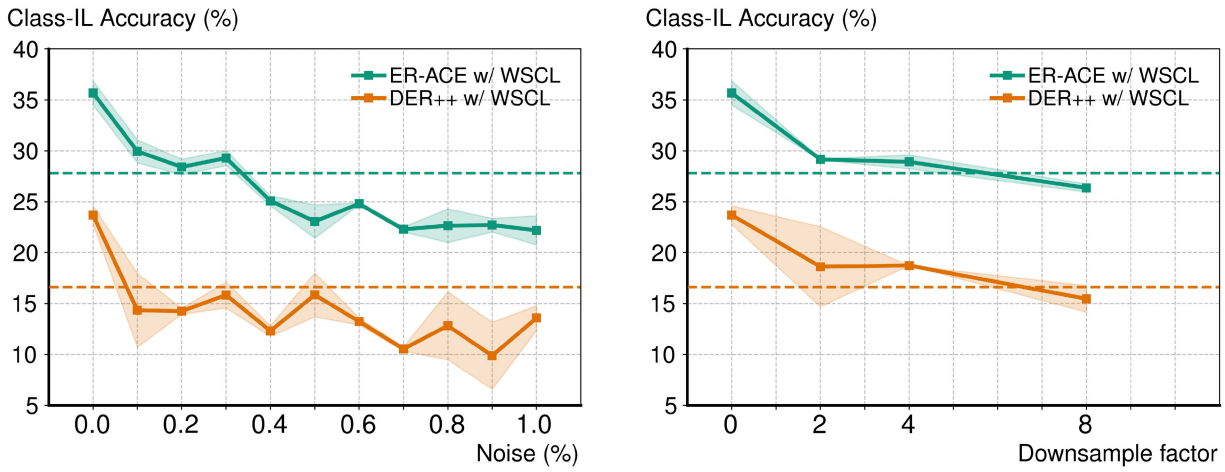


Fig. 2. Impact of dreaming quality, in terms of noise (left) and image resolution (right). Results refer to ER-ACE and DER++ with WSCL (solid lines) and without it (dotted line).

TABLE VIII

COMPARISON IN TERMS OF ACCURACY (FAA) AND PARAMETER UPDATES (Δ**U** DEFINED AS THE CHANGE IN PERCENTAGE OF PARAMETERS UPDATES WHEN SELECTIVE FREEZING IS ENABLED) WITH AND WITHOUT SELECTIVE FREEZING OF THE ER-ACE + WSCL APPROACH ON THE TINY-IMAGENET$_{1/2}$ DATASET WITH BUFFER 500

Backbone	**ResNet-18**						**ResNet-151**					
# epochs	**10**		**50**		**100**		**10**		**50**		**100**	
	FAA (↑)	**ΔU** (↓)	**FAA** (↑)	**ΔU** (↓)	**FAA** (↑)	**ΔU** (↓)	**FAA** (↑)	**ΔU** (↓)	**FAA** (↑)	**ΔU** (↓)	**FAA** (↑)	**ΔU** (↓)
Selective Freezing	41.25	+8.83	44.50	-16.22	45.22	-17.14	35.96	+44.73	44.66	-16.43	43.68	-24.07
No Freezing	41.22		44.24		44.26		38.40		45.36		42.62	

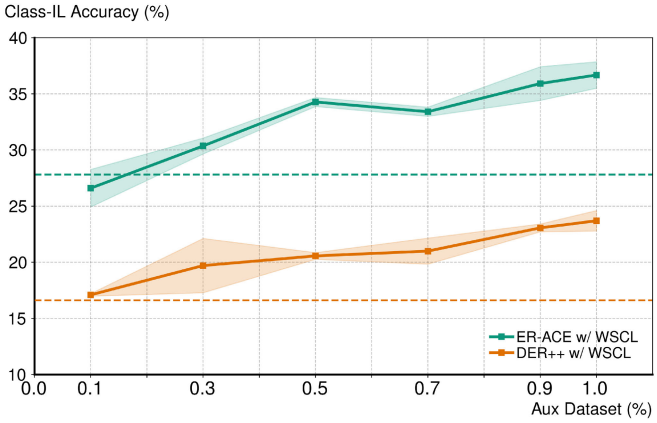


Fig. 3. Impact of dreaming dataset dimension. Results refer to ER-ACE and DER++ with WSCL (solid lines) and without it (dotted line).

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 [57], [58]. 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 with cognitive neuroscience evidence that synchronization of neural activity across different brain regions and changes in the balance between excitation and inhibition enables efficient processing [59], [60].

Fig. 4 shows the most frequent (over ten different runs) set of frozen backbone layers at each task, when training ER-ACE with WSCL on Tiny-ImageNet$_{1/2}$, as well as the total number of performed parameter updates using the training procedure presented in Section IV-B. It is important to note that, however, the freezing strategy employed during the wake stage of WSCL depends on the specific CL method and the target dataset. Fig. 5 illustrates the predominant freezing scheme of ER-ACE when evaluated on the FG-ImageNet dataset, as well as the resulting efficiency. Unlike Tiny-ImageNet$_{1/2}$, where ER-ACE typically freezes almost all layers after the completion of the first task, on FG-ImageNet, ER-ACE gradually freezes the layers of its backbone network until Task τ_5. Despite this more gradual freezing strategy, the efficiency gain achieved is approximately 17.14%, indicating fewer updates compared to the baseline model trained without the wake-sleep strategy in WSCL. Thus, 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% fewer updates), thus confirming the suitability of the wake stage in supporting efficient training.

Furthermore, we conducted an analysis to assess how the freezing strategy scales with the backbone size and the number of epochs, focusing on the performance of ER-ACE on the Tiny-ImageNet$_{1/2}$ dataset. Our results, reported in Table VIII, indicate that the efficiency gains achieved through selective freezing tend to increase with both the number of epochs

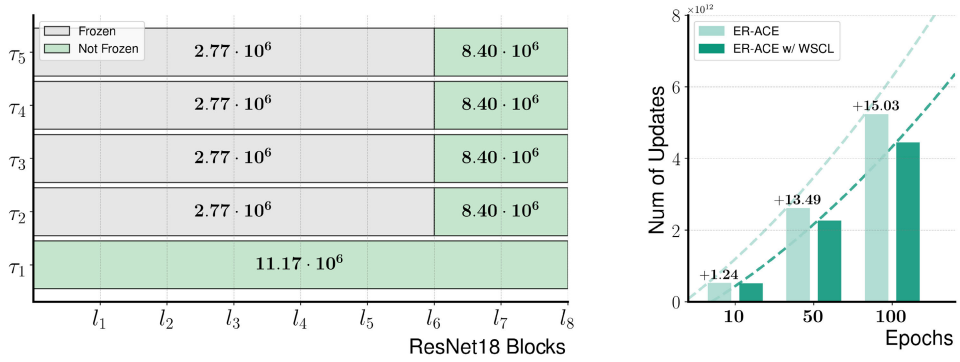


Fig. 4. WSCL model efficiency. The most frequent automatically learned freezing scheme (values within bars are a number of parameters) during the wake phase for ER-ACE on Tiny-ImageNet$_{1/2}$ (left). The numbers above the green bars represent the improvement in percent points with respect to the baseline alone. The 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) (right).

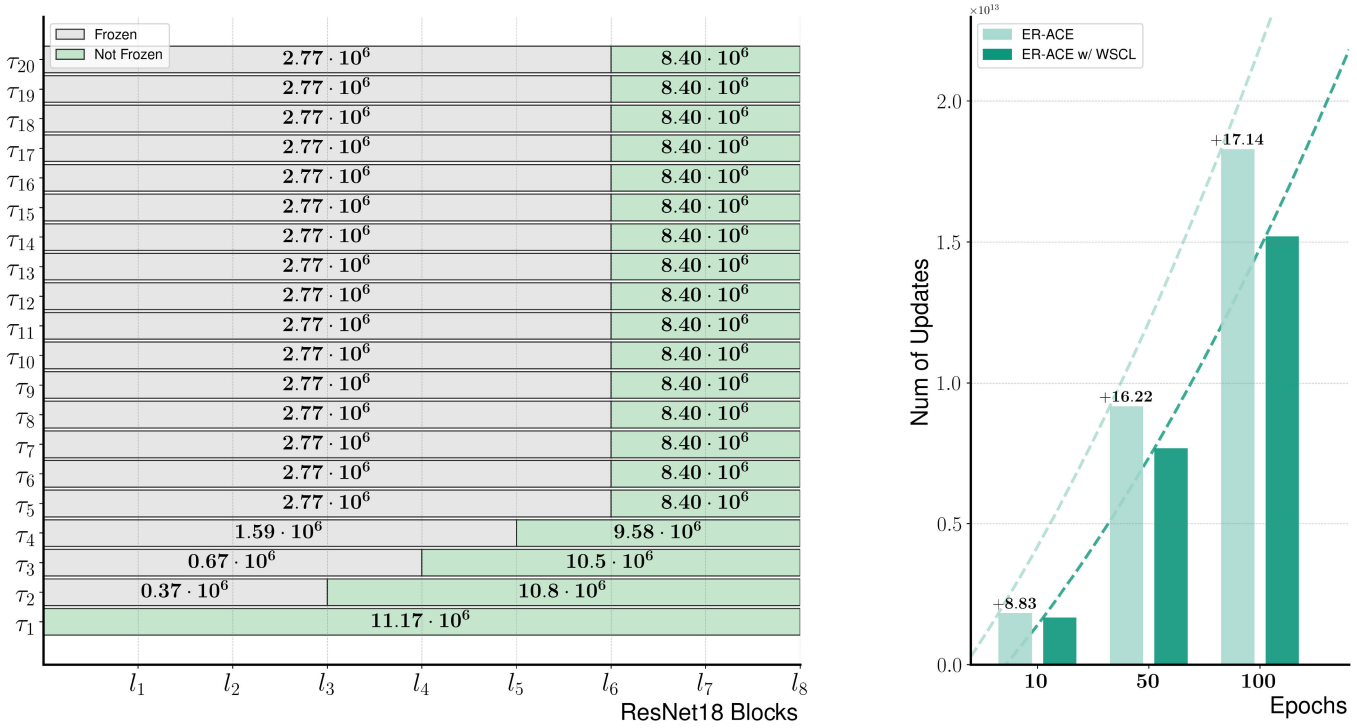


Fig. 5. WSCL model efficiency. The most frequent automatically learned freezing scheme (values within bars are a number of parameters) during the wake phase for ER-ACE on FG-ImageNet (left). The numbers above the green bars represent the improvement in percentage points with respect to the baseline alone. The number of parameter updates for the whole training of ER-ACE with and without WSCL on FG-ImageNet (from 10 epochs to 100 training epochs) (right).

and the depth of the considered model. For instance, when training ResNet18 for 100 epochs, we observed a reduction of approximately 17% in parameter updates compared to the baseline model. In contrast, with ResNet151 and the same number of epochs, the reduction in parameter updates reached about 24%. Notably, the selective freezing strategy primarily targets reducing training computation costs and has minimal impact on performance during inference. These findings underscore the scalability and effectiveness of the wake stage in WSCL for achieving efficient CL across a range of model architectures and training durations.

V. Conclusion

The integration of CLSs theory and sleep mechanisms in artificial neural networks holds great potential for enhancing CL capabilities. Inspired by the interaction between the hippocampus and neocortex in humans, WSCL introduces a sleep phase that mimics off-line brain states during which memory consolidation and synaptic reorganization occur. 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 FWT, 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.

ACKNOWLEDGMENT

Matteo Pennisi and Amelia Sorrenti are Ph.D. students enrolled in the National Ph.D. in Artificial Intelligence, cycle XXXVII and XXXVIII, respectively, the course on Health and Life Sciences, organized by University Campus Bio-Medico of Rome.

REFERENCES

[1] D. Kumaran, D. Hassabis, and J. L. McClelland, "What learning systems do intelligent agents need? Complementary learning systems theory updated," *Trends Cognit. Sci.*, vol. 20, no. 7, pp. 512–534, Jul. 2016.

[2] 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, no. 3, pp. 419–457, Jul. 1995.

[3] Q. Pham, C. Liu, and S. Hoi, "DualNet: Continual learning, fast and slow," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 16131–16144.

[4] Z. Wang et al., "DualPrompt: Complementary prompting for rehearsal-free continual learning," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 631–648.

[5] D. Ji and M. A. Wilson, "Coordinated memory replay in the visual cortex and hippocampus during sleep," *Nature Neurosci.*, vol. 10, no. 1, pp. 100–107, Jan. 2007.

[6] M. P. Walker and R. Stickgold, "Sleep-dependent learning and memory consolidation," *Neuron*, vol. 44, no. 1, pp. 121–133, Sep. 2004.

[7] D. Singh, K. Norman, and A. Schapiro, "A model of autonomous interactions between hippocampus and neocortex driving sleep-dependent memory consolidation," *Proc. Nat. Acad. Sci. USA*, vol. 119, no. 44, Nov. 2022, Art. no. e2123432119.

[8] M. Steriade, D. A. McCormick, and T. J. Sejnowski, "Thalamocortical oscillations in the sleeping and aroused brain," *Science*, vol. 262, no. 5134, pp. 679–685, Oct. 1993.

[9] G. P. Krishnan et al., "Cellular and neurochemical basis of sleep stages in the thalamocortical network," *eLife*, vol. 5, Nov. 2016, Art. no. e18607.

[10] N. Deperrois, M. A. Petrovici, W. Senn, and J. Jordan, "Learning cortical representations through perturbed and adversarial dreaming," *eLife*, vol. 11, Apr. 2022, Art. no. e76384.

[11] M. De Lange et al., "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, Jul. 2022.

[12] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Netw.*, vol. 113, pp. 54–71, May 2019.

[13] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychol. Learn. Motivat.*, vol. 24, pp. 109–165, Jan. 1989.

[14] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, 2017.

[15] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3987–3995.

[16] J. Schwarz et al., "Progress & compress: A scalable framework for continual learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4528–4537.

[17] A. Mallya and S. Lazebnik, "PackNet: Adding multiple tasks to a single network by iterative pruning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7765–7773.

[18] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Sci.*, vol. 7, no. 2, pp. 123–146, Jun. 1995.

[19] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "ICaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5533–5542.

[20] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: A strong, simple baseline," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 15920–15930.

[21] R. Aljundi, K. Kelchtermans, and T. Tuytelaars, "Task-free continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11246–11255.

[22] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias, "Three types of incremental learning," *Nature Mach. Intell.*, vol. 4, no. 12, pp. 1185–1197, Dec. 2022.

[23] R. Ratcliff, "Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions," *Psychol. Rev.*, vol. 97, no. 2, pp. 285–308, 1990.

[24] P. Dokania, P. Torr, and M. Ranzato, "Continual learning with tiny episodic memories," in *Proc. Workshop Multi-Task Lifelong Reinforcement Learn.*, 2019.

[25] E. Frascaroli et al., "Latent spectral regularization for continual learning," 2023, *arXiv:2301.03345*.

[26] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 11817–11826.

[27] A. Chaudhry, A. Gordo, P. Dokania, P. Torr, and D. Lopez-Paz, "Using hindsight to anchor past knowledge in continual learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 6993–7001.

[28] M. D. Lange and T. Tuytelaars, "Continual prototype evolution: Learning online from non-stationary data streams," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8250–8259.

[29] 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 *Proc. Int. Conf. Learn. Represent.*, 2022. [Online]. Available: https://openreview.net/forum?id=N8MaByOzUfb

[30] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 3584–3594.

[31] Y. Guo, B. Liu, and D. Zhao, "Online continual learning through mutual information maximization," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 8109–8126.

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

[33] H. Cha, J. Lee, and J. Shin, "Co2L: Contrastive continual learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9496–9505.

[34] E. Arani, F. Sarfraz, and B. Zonooz, "Learning fast, learning slow: A general continual learning method based on complementary learning system," in *Proc. Int. Conf. Learn. Represent.*, 2022. [Online]. Available: https://openreview.net/forum?id=uxxFrDwrE7Y

[35] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, "The 'wake-sleep' algorithm for unsupervised neural networks," *Science*, vol. 268, no. 5214, pp. 1158–1161, May 1995.

[36] J. Bornschein and Y. Bengio, "Reweighted wake-sleep," 2014, *arXiv:1406.2751*.

[37] T. Tadros, G. P. Krishnan, R. Ramyaa, and M. Bazhenov, "Sleep-like unsupervised replay reduces catastrophic forgetting in artificial neural networks," *Nature Commun.*, vol. 13, no. 1, p. 7742, Dec. 2022.

[38] M. Y. Harun, J. Gallardo, T. L. Hayes, R. Kemker, and C. Kanan, "SIESTA: Efficient online continual learning with sleep," *Trans. Mach. Learn. Res.*, 2023.

[39] S. Golkar, M. Kagan, and K. Cho, "Continual learning via neural pruning," 2019, *arXiv:1903.04476*.

[40] M. Wortsman et al., "Supermasks in superposition," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 15173–15184.

[41] J. Pomponi, S. Scardapane, and A. Uncini, "Structured ensembles: An approach to reduce the memory footprint of ensemble methods," *Neural Netw.*, vol. 144, pp. 407–418, Dec. 2021.

[42] M. J. Fosse, R. Fosse, J. A. Hobson, and R. J. Stickgold, "Dreaming and episodic memory: A functional dissociation?" *J. Cognit. Neurosci.*, vol. 15, no. 1, pp. 1–9, Jan. 2003.

[43] S. Llewellyn, "Dream to predict? REM dreaming as prospective coding," *Frontiers Psychol.*, vol. 6, p. 1961, Jan. 2016.

[44] S. Schwartz, "Are life episodes replayed during dreaming?" *Trends Cognit. Sci.*, vol. 7, no. 8, pp. 325–327, Aug. 2003.

[45] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6470–6479.

[46] G. Bellitto, M. Pennisi, S. Palazzo, L. Bonicelli, M. Boschini, and S. Calderara, "Effects of auxiliary knowledge on continual learning," in *Proc. 26th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2022, pp. 1357–1363.

[47] Stanford. (2015). *Tiny ImageNet Challenge (CS231n)*. [Online]. Available: https://www.kaggle.com/c/tiny-imagenet.

[48] A. Prabhu, P. H. Torr, and P. K. Dokania, "GDumb: A simple approach that questions our progress in continual learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 524–540.

[49] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[50] V. Lomonaco and D. Maltoni, "Core50: A new dataset and benchmark for continuous object recognition," in *Proc. Conf. Robot Learn.*, 2017, pp. 17–26. [Online]. Available: https://proceedings.mlr.press/v78/lomonaco17a.html

[51] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[52] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," in *Proc. Int. Conf. Learn. Represent.*, 2019. [Online]. Available: https://openreview.net/forum?id=Hkf2_sC5FX

[53] Y. Wu et al., "Large scale incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 374–382.

[54] A. Benjamin, D. Rolnick, and K. Kording, "Measuring and regularizing networks in function space," in *Proc. Int. Conf. Learn. Represent.*, 2019. [Online]. Available: https://openreview.net/forum?id=SkMwpiR9Y7

[55] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 831–839.

[56] F. Pernici, M. Bruni, C. Baecchi, F. Turchini, and A. Del Bimbo, "Class-incremental learning with pre-allocated fixed classifiers," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6259–6266.

[57] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-inspired artificial intelligence," *Neuron*, vol. 95, no. 2, pp. 245–258, Jul. 2017.

[58] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Netw.*, vol. 111, pp. 47–63, Mar. 2019.

[59] 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, Jan. 2014.

[60] L. Marshall and J. Born, "The contribution of sleep to hippocampus-dependent memory consolidation," *Trends Cognit. Sci.*, vol. 11, no. 10, pp. 442–450, Oct. 2007.

Amelia Sorrenti received the master's degrees in computer engineering from the University of Catania, Catania, Italy, in 2022, where she is currently pursuing the Ph.D. degree in artificial intelligence with the University Campus Bio-Medico of Rome, Rome, Italy.

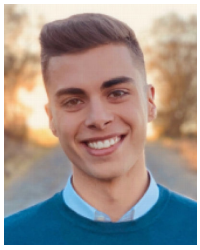
She is a member with the PeRCeiVe Lab Research Group, University of Catania. Her research is primarily focused on bio-inspired machine learning, incremental, and continual learning.

Giovanni Bellitto (Member, IEEE) received the Ph.D. degree from the University of Catania, Catania, Italy, in 2023.

He has been an Assistant Professor at the University of Catania since 2024. He has been a Member with the PeRCeiVe Lab Research Group since 2019. His research interests include machine learning and artificial intelligence, with a particular focus on bio-inspired methods for incremental learning, federated learning, and medical image analysis.

Federica Proietto Salanitri (Member, IEEE) received the Ph.D. degree from the University of Catania, Catania, Italy, in 2023.

Since January 2023, she has been an Assistant Professor at the Department of Electrical Electronic and Computer Engineering, University of Catania, where she is a Member with the PeRCeiVe Lab Research Group. Her research interests include machine learning and artificial intelligence, with a particular focus on medical image analysis, federated learning, explainable AI, and computer vision in a broader context.



Matteo Pennisi is currently pursuing the dual Ph.D. degree with the University of Catania, Catania, Italy, and the University Campus Bio-Medico of Rome, Rome, Italy, with the National AI Ph.D. Program (Health Pillar).

His research interests include mainly generative models, continual learning, and federated learning with applications in general deep learning and medical imaging.

Simone Palazzo received the Ph.D. degree from the Department of Electrical, Electronic and Computer Engineering, University of Catania, Catania, Italy, in 2018.

He is currently an Assistant Professor at the Department of Electrical, Electronic and Computer Engineering, University of Catania. His research interests include machine learning, artificial intelligence, and pattern recognition, with a particular focus on computer vision, medical imaging, and explainable AI and bio-inspired methods for machine learning.

Concetto Spampinato (Member, IEEE) received the Ph.D. degree in computer engineering at the University of Catania, Catania, Italy, in 2008.

He specializes in artificial intelligence and machine learning, with a focus on AI models driven by cognitive neuroscience principles. He is currently a Professor of computer engineering with the University of Catania, Catania, Italy, and a Courtesy Faculty Member at the Center for Research in Computer Vision, University of Central Florida, Orlando, FL, USA. His research interests include computer vision, continual and incremental learning, medical image analysis, and AI for robotics.