Realistic Continual Learning Approach using Pre-trained Models

Nadia Nasri[a,b], Carlos Gutiérrez-Álvarez[a], Sergio Lafuente-Arroyo[a], Saturnino Maldonado-Bascón[a], Roberto J. López-Sastre[a]

[a] *University of Alcalá, Department of Signal Theory and Communications, GRAM Research group, Alcalá de Henares, 28805, Spain*
[b] *University of Alicante, Department of Computer Science and Artificial Intelligence, San Vicente del Raspeig, 03690, Spain*

Abstract

Continual learning (CL) evaluates adaptability in learning solutions to retain knowledge. Our research addresses the challenge of catastrophic forgetting, where models lose proficiency in previously learned tasks as they acquire new ones. While numerous solutions have been proposed, existing experimental setups often rely on idealized class-incremental learning scenarios. We introduce Realistic Continual Learning (RealCL), a novel CL paradigm where class distributions across tasks are random. We also present CLARE (Continual Learning Approach with pRE-trained models for RealCL scenarios), a pre-trained model-based solution designed to integrate new knowledge while preserving past learning. Our contributions include pioneering RealCL as a generalization of traditional CL setups, proposing CLARE as an adaptable approach for RealCL tasks, and conducting extensive experiments demonstrating its effectiveness across various RealCL scenarios. Notably, CLARE outperforms existing models on RealCL benchmarks, highlighting its versatility in unpredictable learning environments. Code to reproduce all our experiments can be found at https://github.com/gramuah/clare.

Keywords: Deep learning, Continual learning, Realistic scenario, Pre-trained models

1. Introduction

Continual learning (CL), also referred to as lifelong or incremental learning, is a vibrant sub-field of machine learning focused on creating algorithms that can learn from sequential input data streams in an incremental manner. Its major purpose is to mimic human beings' ability to continuously gather and refine new information over time, allowing models to adapt to new information without sacrificing the previously learned one. The implication of CL is built on the notion that artificial intelligence (AI) should become more robust and adaptable in an ever-changing world. In other words, defining an experimental setup for continual learning requires structuring learning into tasks and evaluating how the accuracy of proposed solutions evolves as tasks progress. Is the model able to maintain good recognition accuracy in the last task it was evaluated on? How has the degradation been in previous tasks when this new task was learned? The main factor driving research in CL is what is known as catastrophic forgetting. It is well known that in a continual learning training scheme based on fine-tuning for deep learning models, for example, when we adjust the model's weights to learn

the last task, there is no guarantee that the model will retain the knowledge it has previously learned (French, 1993; McCloskey and Cohen, 1989). What we observe is called catastrophic forgetting, as the performance in tasks where the model was previously an expert gradually decreases.

Research in CL aims to mitigate the impact of this catastrophic forgetting. Numerous solutions have been proposed based on different principles (e.g. (Shi et al., 2023; Aljundi et al., 2019; Buzzega et al., 2020; Shim et al., 2020; Zhao et al., 2020; Castro et al., 2018; Wang et al., 2023; Douillard et al., 2021; Zhou et al., 2023; Zhao et al., 2019; Li and Hoiem, 2016; Zhou et al., 2019; Douillard et al., 2020)), but they all share an aspect that in our opinion has gone unnoticed: the idealized or unrealistic experimental setup. To compare different continual learning models, researchers have defined different experimental setups in which we simulate a continuous learning process by controlling the distribution of data into different tasks. We take a specific dataset, such as CIFAR-100, and organize the different object categories into tasks, so we ask our continual learning models to gradually learn to recognize, for example, between the categories dog and cat first. Then in a second task between the categories car, motorcycle, dog and cat. And so on. This is what is called a class-incremental learning scenario. The models proposed in the literature present ad hoc solutions for this experimental situation. In other words, the models have been specifically designed to optimize their learning when categories grow in a structured and incremental manner. The question we ask from this work is: what happens when this assumption is not met? Imagine a realistic continual learning environment where these idealizations do not exist. The data that the continual learning model has to face is also structured into tasks, but the distribution of categories is not controlled or structured but randomly formed. Categories like dog and cat may appear in task 1, but they may also reappear later in task n. In this work, we propose to advance in the problem of continual learning by focusing on how state-of-the-art models have benefited from this idealization of the experimental setup. Therefore, we propose a generalization of the problem that we have called Realistic CL (RealCL), in which CL models will have to deal with task streams where the class distribution is not structured (see Figure 1).

To explore this new RealCL scenario, we also introduce a novel solution for continual learning that exploits the benefits of using what is known as pre-trained models (e.g. (Radford et al., 2021; Dosovitskiy et al., 2021)). Our model is named CLARE, for a Continual Learning Approach with pRE-trained models for RealCL scenarios. Distinguished by its simplicity and pre-trained-model-agnostic nature, CLARE leverages pre-trained models to seamlessly integrate new knowledge while preserving past learning. This breakthrough eliminates the need for extensive retraining learning process. In summary, our main contributions are three-fold:

- We pioneer the challenging task of RealCL, a new continual learning setup where the class distribution is not structured across tasks. We proof that RealCL is the generalization of all the standard CL scenarios typically employed in all previous scientific works. In our study we explore a spectrum of CL scenarios, encompassing highly structured to entirely unstructured. Our objective is to reveal whether the state-of-the-art CL models have taken advantage of the artificial and forced structuring that characterizes traditional experimental evaluation scenarios, evaluating them in the novel RealCL scenario.

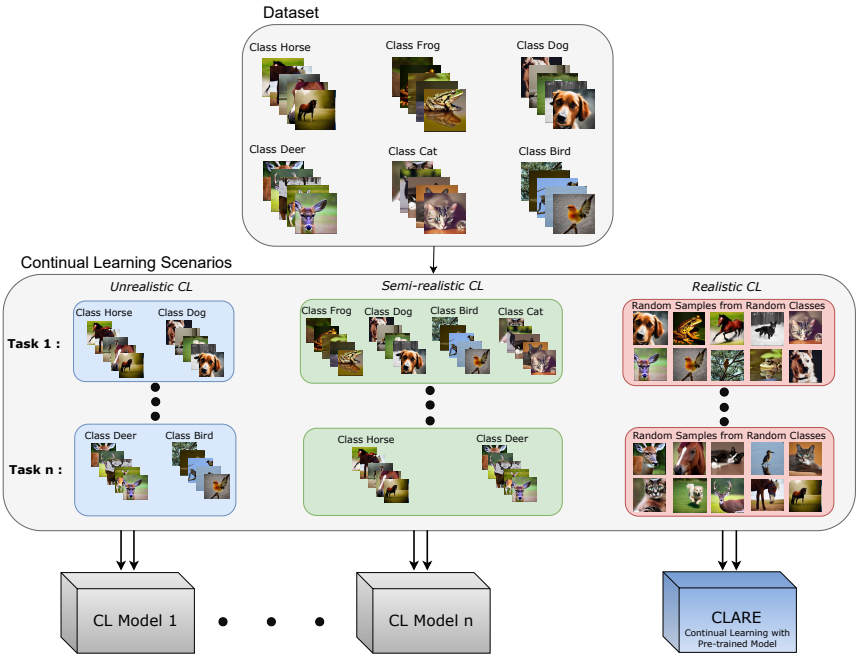


Figure 1: In this figure, we illustrate the possible experimental evaluation scenarios that can be used to validate continual learning solutions. The scenario referred to as Unrealistic CL is the most commonly used in the literature, also known as the class-incremental setup. In this work, we propose to explore the RealCL scenario as a generalization of all continual learning setups. In RealCL, the categories to be recognized by CL models are not distributed in a structured manner across tasks, but rather randomly. Halfway between Unrealistic CL and RealCL, we have the semi-realistic scenario, where it is tolerated for classes to be organized across tasks, but the number classes assigned to each task is not controlled. In this work, we provide an experimental evaluation in the new RealCL scenario of both our novel CLARE proposal and the state-of-the-art CL models.

- We propose the pre-trained model based continual learning strategy CLARE, which is an innovative and adaptable approach tailored for RealCL tasks. CLARE is based on the ability that pre-trained models seem to exhibit to capture and retain knowledge when combined with a specific dynamic module on which learning is performed.
- Extensive experiments are conducted to verify the effectiveness of our CLARE model in all the continual learning scenarios that can be derived from RealCL. Impressively, our method achieves a strong performance even improving some state-of-the-art models when tested on the RealCL benchmarks. Through meticulous comparisons, we assess CLARE's performance against alternative approaches, demonstrating its versatility and robustness in handling the unpredictability imposed by the RealCL setup, which simulates what a realistic CL scenario would be like, where we have no control over the tasks that the models are being confronted with.

2. Related Work

2.1. Continual learning and its idealized evaluation scenarios

Continual learning is a sub-field of machine learning concerned with how models can be trained on a stream of data arriving sequentially without performance decay

or forgetting. This requires solving the stability-plasticity trade-off, which is a core problem in CL (Mermillod et al., 2013; Lee et al., 2022; Araujo et al., 2022). This trade-off refers to how a model should balance retaining old knowledge (stability) and adding new knowledge (plasticity). Therefore, it is a problem of catastrophic forgetting that stems from the necessity to combine two fairly outstanding strengths – stability and plasticity (French, 1993; McCloskey and Cohen, 1989). Where stability makes sure a model holds on to the knowledge it previously gained and plasticity allows it to integrate new data. However, quantifying the importance of these two characteristics and finding an algorithm that accounts for both puts the upfront rule in this task. Usually, the most recent data takes the most weight, eroding previous insights, which is a phenomenon of catastrophic forgetting. It is most fatal in tasks that need to adjust to ever-changing datasets and new contexts. Given this obstacle, research in the area of CL has shown certain ways that are trying to reduce catastrophic forgetting, so as not to lose valuable knowledge and transfer it between different tasks.

Three main techniques are employed in the realm of CL to address the issue of catastrophic forgetting. The first group relies on replaying old samples or exemplars from previous classes, and it is commonly known as rehearsal (Aljundi et al., 2019; Buzzega et al., 2020; Shim et al., 2020; Zhao et al., 2020; Castro et al., 2018). In this method, data from previous tasks is sampled based on some strategy and fed to the model, which is then again integrated with new data during the training process. Rehearsal was initially presented as a prevalent and most effective approach in CL literature due to its potential to retain diversity and representation of data across tasks and avoid the forgetting of previous knowledge (Lomonaco et al., 2020).

The second group addresses the challenge of limited representation capacity by dynamically expanding the network architecture upon the introduction of new tasks (Wang et al., 2023; Douillard et al., 2021; Zhou et al., 2023). In this case, it is ensured by network self-growing in a dynamic way upon each arriving task. Techniques involving the addition of new neurons, layers, or modules enable this smooth integration of new information without disturbing previously acquired knowledge. The improvement of the feature space expressiveness and its diversity makes these kinds of expansions reduce the model parameter interference and saturation. Some well-known examples include progressive neural networks, dynamic expandable networks, and lifelong learning via network growth domains.

The final category of CL strategies is based on knowledge distillation, where the objective is to facilitate knowledge extraction from old models and transfer it into new ones (Zhao et al., 2019; Li and Hoiem, 2016; Zhou et al., 2019; Douillard et al., 2020). This technique ensures that the transfer of knowledge from a larger model to a smaller model does not compromise its validity. In terms of CL, knowledge distillation makes sure that outputs of older models are aligned with those of newer ones such that output probabilities in the new model match those in the old model when subjected to the same input. This synchronization guarantees the retention of knowledge pertaining to earlier concepts, thereby preventing forgetting during the acquisition of new tasks. It is important to acknowledge that some of the studies do not fit to one type of CL methods since they leverage different techniques in combination for getting better outcomes.

Most of the above-mentioned works were conducted within the context of well-defined experimental scenarios, where the classes were also balanced across the tasks. Although such experiments are indispensable for studying the catastrophic forgetting

phenomenon, they often fail because real-world data streams are dynamic and non-stationary. Considering the probability of some classes reappearing, it implies that class distributions and their frequencies in these data streams can dynamically alter from task to task in ways that could seem unexpected for the model.

In other studies, there have been several other initiatives with differing settings of continual learning but no less important; which strongly reflect the value of proper realistic evaluation conditions (Chen et al., 2019; van de Ven and Tolias, 2019; De Lange and Tuytelaars, 2021; van de Ven et al., 2022). Yet, it is worth noting that the data distribution in these scenarios often remains fixed or controlled, stipulated with class frequencies for the task. The diversities among these scenarios are essential because they reveal different perspectives on how continual learning can be done in real-world situations.

In our work, we present a new Realistic Continual Learning (RealCL) scenario, carefully designed to learn from a non-stationary series of tasks. This scenario is unique because it does not control the distribution of classes across tasks or the recurrence of a class. Our approach is intentionally crafted to reflect the natural distribution of the data stream, capturing the diversity and complexity found in real-world situations.

2.2. Pre-trained Models for CL

Recently, high-performance and flexible pre-trained models (PTMs) have been widely adopted as backbones for continual learning. In particular, the Vision Transformer (ViT) (Dosovitskiy et al., 2021) has become one of the most popular architectures due to its strong representation capabilities. The original ViT architecture processes an image as a sequence of embedded image patches using a standard transformer encoder and contains only the parameters associated with the patch embedding, positional embeddings, transformer blocks, and classification token.

Building upon this architecture, several prompt-based continual learning methods introduce a small set of additional learnable prompt parameters that are optimized while keeping the pre-trained ViT backbone frozen or largely unchanged. These prompts act as task-specific conditioning tokens, enabling the model to retrieve task-relevant knowledge while mitigating catastrophic forgetting and promoting knowledge transfer across tasks. Representative methods include L2P (Wang et al., 2021), DualPrompt (Wang et al., 2022c), S-Prompt (Wang et al., 2022a), DAP (Jung et al., 2023), CODA-Prompt (Smith et al., 2022), and other related approaches (Wang et al., 2022b). In some cases, prompt learning has also been extended to multimodal continual learning scenarios by jointly learning image and text prompts to better exploit complementary visual and semantic information (Wang et al., 2022a; Zhou et al., 2021).

Recent advances have taken advantage of the generalizable embeddings of PTMs to address the CL challenges and propose an innovative framework, AdaPt and mERge (APER) (Zhou et al., 2024), that significantly outperforms traditional methods without requiring downstream task training. It does this by combining the adaptivity of model updating with the generalizability of knowledge transfer.

Different methodologies, in the alternative direction, tend to have very low parameter costs while they also allow effective adaptation of PTMs to downstream tasks through focusing the fine-tuning process on a few parameters. In particular, approaches like Visual Prompt Tuning (VPT) (Jia et al., 2022) add prefix tokens that are tunable, and Low-Rank Adaptation (LoRA) (Hu et al., 2021), which employs low-rank matrices

in updating the parameters, are methods that can achieve close to the fine-tuned model accuracy at negligible computational costs.

These methods, however, have their limitations despite the progress made. In most cases, prompts have few static dimensions and a simplistic way of selecting them. On the other hand, the performance of a tuning method, which is also parameter-efficient, has to balance with the number of parameters it can tune. But still, this integration process did turn out to be very well-working because it enables CL to work with PTMs and they both provide this potential for knowledge preservation; as such models outperform scratch-trained models and open new vistas for research in this area (Cao et al., 2024).

Building on the progress of PTMs, in this paper we introduce CLARE, a novel model that is able to exploit the powerful representation of a frozen PTM, combined with a Dynamic Neural Adaptation Network (Dyn-NAN) module. We do not follow any prompting-based pipeline, but propose with CLARE a simple and effective architecture that exhibits a remarkable performance in the realistic CL setting proposed in this work, as well as in the traditional CL scenarios. Unlike standard replay-based methods, CLARE introduces this novel combination of a frozen pre-trained model with a Dyn-NAN. This architecture enables efficient task adaptation without retraining the pre-trained model, focusing instead on dynamically adjusting the Dyn-NAN module to integrate new knowledge while minimizing interference with previously learned tasks. By leveraging the strong representations of pre-trained models, CLARE not only mitigates forgetting but also achieves robustness in unstructured and realistic continual learning scenarios, distinguishing it from traditional replay approaches.

3. From Unrealistic to Realistic Continual Learning

In this section we detail the novel realistic continual learning setting we aim to explore in this work. We demonstrate that the standard scenarios typically employed in all previous scientific works function as mere specializations of our realistic scenario, given its broader scope, which encompasses them all.

3.1. Realistic Continual Learning

Realistic continual learning (RealCL) aims to learn from an evolving task stream where no control over the distribution of new classes in the new tasks is applied. The objective of a RealCL approach is to learn a model for a large number of tasks sequentially, without forgetting knowledge obtained from the preceding tasks, where the data in the old tasks are not available anymore during training new ones. We propose a training scheme that does not impose *any* control over how the data stream is distributed across different tasks, as it gradually arrives. Neither the classes nor the distribution of samples arriving with each task are controlled. Note that this is different from a naive fine-tuning based training process in CL, where the new tasks provide access to new categories, typically balancing the number of features and the class distributions. In our RealCL, all the task definition process is random. We assume, therefore, that the data flow randomly into our system, and once the data has been used for learning, our model cannot access it as new tasks arrive.

We now formalize the mathematical problem for RealCL, which will allow us, in Section 3.2, to proof how traditional scenarios used in CL are particularizations of our proposal. In a RealCL setting we assume that there is a sequence of K training

tasks $\{\mathcal{T}^1, \mathcal{T}^2, \cdots, \mathcal{T}^K\}$, where $\mathcal{T}^k = \left\{\left(\mathbf{x}_i^k, y_i^k\right)\right\}_{i=1}^{n_k}$ is the k-th incremental step with n_k instances. Here, the training instance $\mathbf{x}_i^k \in \mathbb{R}^D$, where D is the dimensionality of the feature space, belongs to class $y_i \in Y_k$, where Y_k is the label space of task k. Note that this label space is random, so there is no control over the labels that should appear in each task. Furthermore, another property of the proposed RealCL paradigm is that the original training data from previous tasks are no longer accessible once a task has been completed. In other words, during incremental step k, the learner cannot revisit the *complete* datasets associated with previous tasks. This assumption reflects realistic deployment scenarios in which historical data cannot be permanently stored or reloaded. Notice that this restriction does not preclude continual learning algorithms from maintaining a limited auxiliary memory (e.g., a replay buffer or a compact set of exemplars), provided that the complete historical datasets remain unavailable.

The aim of RealCL is to incrementally construct a unified model encompassing all observed classes, i. e. assimilating information from new classes while simultaneously retaining knowledge from previous ones. During training, the learner sequentially updates the model using only the current training task $\mathcal{T}^k$, without accessing previous training data. The learning objective at each incremental step is therefore defined exclusively over the available training set. Overall, the goal is to learn a model $f(\mathbf{x}) : \mathbb{R}^D \to \mathcal{Y}_k$ using only the training data available at each incremental step.

After completing task k, the model is evaluated on the union of the *testing* sets corresponding to all observed tasks in order to assess both knowledge acquisition and forgetting, i.e. $\mathcal{T}_{\text{test}}^1 \cup \cdots \mathcal{T}_{\text{test}}^k$.

For the evaluation metric we use the empirical risk,

$$\sum_{(\mathbf{x}_j, y_j) \in \mathcal{T}_{\text{test}}^1 \cup \cdots \mathcal{T}_{\text{test}}^k} \ell\left(f\left(\mathbf{x}_j\right), y_j\right), \tag{1}$$

where $\ell(\cdot, \cdot)$ measures the discrepancy between prediction and ground-truth labels, and $\mathcal{T}_{\text{test}}^k$ encodes the *testing* samples for task k. A well-performing RealCL model that meets the requirements of Eq. 1 exhibits discriminability across all classes, achieving a harmonious equilibrium between learning novel classes and retaining memory of existing ones.

3.2. The unrealistic scenarios

Traditionally, in all scientific publications, the problem of continual learning has been addressed through experimental evaluations that are particularizations of the RealCL model. These particularizations arise from imposing some restrictions on the more general RealCL model in different aspects, which we detail as follows.

Most continual learning benchmarks proposed in the literature rely on what is commonly referred to as the non-overlapping tasks paradigm. In these settings, the label space is explicitly partitioned into disjoint subsets, each assigned to a different task. Consequently, once a class has appeared in one task, it never reappears in subsequent tasks. Unlike these formulations, RealCL imposes no constraints on the assignment of classes to tasks. Since tasks are generated directly from the incoming data stream, previously observed classes may naturally reappear in later tasks, while entirely new classes may also emerge. Therefore, the non-overlapping tasks paradigm can be regarded as a *particular restriction* of the more general RealCL formulation.

Therefore, in the non-overlapping setup, the label space is partitioned into distinct, non-overlapping subsets, each corresponding to a task, and the abrupt shifts between these subsets are denoted as task boundaries. While in the RealCL scenario there is no control over the balancing of categories or their distribution in each task, in the traditional non-overlapping tasks setting there is. Typically, the task boundaries delineate data splits that are balanced, ensuring an approximately equal distribution of categories. This formulation substantially simplifies the overarching continual learning problem, constraining the previously unknown expansion of the label space and rendering it comprehensible. Technically, the traditional non-overlapping tasks scenario can be viewed as a constrained version of RealCL, obtained by enforcing pairwise-disjoint label spaces Y_k and controlled task composition.

We can further restrict the non-overlapping tasks scenario by assuming that the categories distributed among the different tasks do not overlap and gradually increase, being balanced across tasks. This is traditionally known in the continual learning literature as the class-incremental scenario (Rebuffi et al., 2017; Shmelkov et al., 2017). The class-incremental scenario is therefore a particular case of the RealCL setting, so that the label spaces of each task $\mathcal{T}^k$ are distributed as follows: $Y_k \cap Y_{k'} = \varnothing$ for $k \neq k'$.

Another unrealistic but popular choice for the continual learning formulation is the task-incremental setting (e.g. (Lopez-Paz and Ranzato, 2017)). In this context, in addition to the non-overlapping and class-incremental task premises, a task descriptor, or task ID, is also conveyed by an oracle both during training and testing phases. In particular, the continual learning model observes for every task $\mathcal{T}^k = \left\{\left(\mathbf{x}_i^k, k, y_i^k\right)\right\}_{i=1}^{n_k}$, where k is the task descriptor. Being aware of this subset of labels a priori significantly reduces the label space during both training and inference, but it is relatively impractical to possess such knowledge in real-world scenarios.

Finally, the scenario known as domain-incremental can be defined (e.g. (Goodfellow et al., 2014)). In this setting, the label space is shared across tasks, that is: $Y_k = Y_{k'}$ for $k \neq k'$. However, data $\mathbf{x}_i^k$ may originate from changing input domains across tasks (e.g., varying imaging conditions, different viewpoints, permutations, etc.).

All previous formulations provided in this section are particularizations of the more general formulation provided for RealCL, an aspect that confirms that all previous CL experimental settings are simplifications of the realistic scenario proposed.

3.2.1. A Semi-realistic scenario

As a middle ground between the unrealistic scenarios and RealCL, we propose a semi-realistic scenario (SemiRealCL) where we enforce RealCL to be of class-incremental nature.

In the SemiRealCL setting, we ensure that once a class is introduced in a task, it does not reappear in subsequent tasks. However, the number of classes per task is not predefined, it is random, leading to an uneven distribution. See Figure 1 for a better understanding of the proposed setting. In contrast, the RealCL setting introduces no constraints on class distribution, allowing for random recurrence of previously seen classes in future tasks. Therefore, since SemiRealCL must be a class-incremental process, once a set of classes is randomly assigned to a task, those categories cannot be used again in future tasks. The main difference from the standard class-incremental setting is that the categories are not evenly distributed across tasks. For example, in a SemiRealCL setting where we define 5 tasks for a dataset with 10 classes, the distri-

bution of classes across tasks is randomized rather than being grouped in pairs. For instance, task 1 might include classes $\{0, 2, 6\}$, task 2 could have $\{5, 8\}$, task 3 might be assigned $\{1\}$, and so on.

4. CLARE: Continual Learning Approach with pre-trained Models

To explore the new RealCL paradigm, we propose a novel solution based on pre-trained models. We present **CLARE**, our **C**ontinual **L**earning **A**pproach with p**RE**-trained models for RealCL scenarios. Pre-trained models inherently possess a degree of generalizability, which can be leveraged and transferred to downstream tasks. In this context, we introduce CLARE, a framework designed to adapt pre-trained models' recognition capabilities for the RealCL setup.

We show CLARE's main components in Figure 2. As can be observed, CLARE is essentially a continual learning model based on a replay buffer, where we combine: a) a memory module $\mathcal{M}$ with its corresponding sample selection mechanism; b) a pre-trained model used as an information extractor; and c) a learning process aimed at minimizing catastrophic forgetting, which integrates a Dynamic Neural Adaptation Network (Dyn-NAN) block into the optimization process.

Note that CLARE is a model designed to operate under the proposed RealCL protocol. Although RealCL assumes that *complete* historical datasets are unavailable after each task, CLARE follows the standard rehearsal-based continual learning strategy of maintaining a small bounded replay buffer $\mathcal{M}$ containing representative samples from previous tasks.

The use of this bounded replay buffer should not be interpreted as contradicting the motivation behind RealCL. The objective of the proposed RealCL scenario is to remove the unrealistic assumption that complete historical datasets remain permanently available. Maintaining a small exemplar memory is a common compromise adopted in rehearsal-based continual learning, balancing performance with practical constraints on storage and computation.

Therefore, the memory module in CLARE is responsible for retaining a subset of training samples from both current and previously encountered tasks. How does it work? We follow a simple and efficient strategy: we randomly sample instances from the classes of the current task $\mathcal{T}^k$ and store them in the memory module $\mathcal{M}$, together with instances that belong to the classes considered in previous tasks $\{\mathcal{T}^{k-1}, \mathcal{T}^{k-2}, \ldots\}$. So, for each task $\mathcal{T}^k$, we randomly sample instances from the current task's data and add them to the memory buffer $\mathcal{M}$. If the buffer has available space, the new samples are appended directly. Otherwise, a replacement strategy is applied, where a portion of the existing samples is randomly replaced by new ones. This process does not enforce explicit class balancing, and the resulting memory content may reflect the inherent class imbalance of the task stream. This design choice aligns with the RealCL paradigm, where class distributions are unstructured and potentially overlapping across tasks. Therefore, our memory module has only one parameter: the size of samples it can store.

It is noteworthy that this parameter can be dynamically changed, allowing CLARE to adapt to the difficulty of the problem dynamically as well. In the experimental evaluation, we have demonstrated the impact that this parameter has on the system's performance. So, the memory represents a subset of the whole data distribution, which

is used by the pre-trained model encoder and the Dyn-NAN module to solve the current task $\mathcal{T}^k$ minimizing the impact over previous tasks.

The second component of CLARE is a pre-trained model used as visual encoder. Note that CLARE is designed to be model agnostic, ensuring seamless integration with any pre-trained model, which enhances its capability to recognize and process a diverse range of classes it has been exposed to. For our implementation and the experimental evaluation, we have opted to integrate in CLARE the Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021) pre-trained model. CLIP learns a common embedding space for natural language and visual concepts. CLIP employs a dual-encoder architecture that comprises two separate encoders: a Transformer model (Vaswani et al., 2017) for text and a CNN or a ViT (Dosovitskiy et al., 2021) for images. The encoders are trained on a large-scale dataset of 400 million image-caption pairs, collected from various internet domains, to maximize the similarity of text and image pairs that are semantically related and minimize the similarity of those that are not. This enables CLIP to acquire transferable representations for both text and images, which can adapt to new data and be applied to downstream tasks such as image classification, captioning, retrieval, and generation. For CLARE we only employ the visual encoder in CLIP, whose output is connected to the learnable Dyn-NAN block.

Finally, CLARE includes the Dyn-NAN module, which consists of fully connected layers set up with full parameterization, integrated in the optimization process in a dynamic way. Technically, the Dyn-NAN is implemented as a dynamic fully-connected neural network. Its design is specifically tailored to address the challenges posed by the RealCL scenario, where the number of classes is unknown and evolves unpredictably across tasks. The key feature of Dyn-NAN is its dynamic adaptability: for each new task $\mathcal{T}^k$, the output layer is dynamically reconfigured to include neurons corresponding to the newly encountered classes, while preserving the weights associated with previously learned ones. This incremental expansion mechanism allows Dyn-NAN to maintain a unified classification head that grows with the label space, without requiring retraining of the frozen pre-trained encoder. For the experiemntal evaluation, we employed an specific architecture that consists of three fully-connected layers with ReLU activations, and the final layer is dynamically resized to match the total number of classes observed up to task k. This design ensures that CLARE can flexibly accommodate new categories while minimizing interference with previously acquired knowledge. Despite its simplicity, Dyn-NAN plays a pivotal role in enabling CLARE to perform effectively under the RealCL setting, where task boundaries and class distributions are unstructured and potentially overlapping. Its dynamic nature allows the model to adapt to evolving classification demands while preserving stability in the learned representations. In the following section, we detail how the Dyn-NAN module is integrated into the training procedure to ensure that new categories are smoothly incorporated into the model, while minimizing the forgetting of previously acquired knowledge.

4.1. Training procedure for CLARE

Following the RealCL setup, for each task $\mathcal{T}^k$, CLARE first updates the memory module $\mathcal{M}$. This memory buffer contains a selection of the current-task $\mathcal{T}^k$ samples together with a subset of representative samples previously stored. For the RealCL image classification problem we are dealing with, these data come in the form of annotated image sets. Then, the images in the memory module $\mathcal{M}$, denoted by I_i^k, are

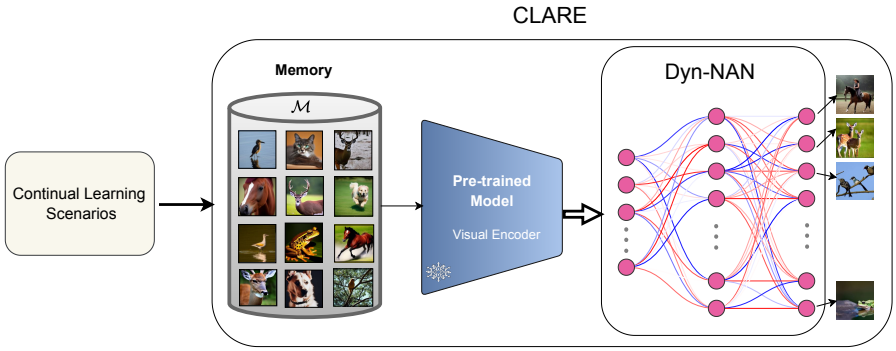


Figure 2: We show the main components of CLARE: the memory buffer $\mathcal{M}$, the frozen pre-trained model that works as the visual encoder, and the Dyn-NAN module.

processed by the frozen visual encoder provided by the pre-trained model, resulting in a mapping from the input images to vectors x_i^k by the encoder E, as stated by the operation $x_i^k \leftarrow E(I_i^k)$.

In other words, to use the pre-trained model involves a mapping from the input images to vectors $\mathbf{x}_i^k$. This way, for task $\mathcal{T}^k$, module Dyn-NAN receives the set $\mathcal{M} = \left\{\left(\mathbf{x}_i^k, y_i^k\right)\right\}_{i=1}^{n_k}$. Dyn-NAN module dynamically adapts the architecture for the possible new number of classes. Dyn-NAN works as a mapping from embeddings $\mathbf{x}_i^k$ to scalar scores for each category, i.e. $\{g_y : \mathbb{R}^d \rightarrow \mathbb{R}\}_{y \in \mathcal{Y}_k}$. The output of Dyn-NAN module is defined as the concatenation of the output heads for all the classes seen so far, i.e. $\mathcal{Y}_k$, as $f_k(\mathbf{x}) : X \rightarrow \mathcal{Y}_k$, where:

$$f_k(\mathbf{x}) = [g_y(\mathbf{x})]_{y \in \mathcal{Y}_k} \,. \tag{2}$$

CLARE model final prediction is obtained as the argmax over the output layer of Dyn-NAN module as follows:

$$\hat{y}_k(\mathbf{x}) = \underset{y \in \mathcal{Y}_k}{\operatorname{argmax}}\, f_k(\mathbf{x})[y] \,. \tag{3}$$

During training, CLARE uses the cross-entropy loss function. The entire training process of CLARE is depicted in Algorithm 1.

Overall, in our training process, we can follow with CLARE two learning strategies. The first one simply involves training all the model's weights from scratch in every task iteration, as the memory module always contains samples from all the classes observes so far. Pre-trained models result effective for such training strategy. The second possibility is to follow a fine-tuning strategy, starting from the weights that the model has from the learning done until the previous task $\mathcal{T}^{k-1}$. Due to the generalization capability of pre-trained models, it is interesting to evaluate the impact of following one strategy or the other. This aspect is shown in the experimental evaluation.

5. Experiments

In this section, we present the experimental evaluation designed for the RealCL problem. We first detail the description of the experimental methodology, highlighting the key factors that influenced our experimental design and execution. Our main goals are to answer the following research questions that enable us to thoroughly evaluate and comprehend the implications of the novel RealCL paradigm:

Algorithm 1 CLARE Training under the RealCL Setting

Require: Memory buffer $\mathcal{M}$ (capacity B), frozen encoder E, task sequence $\{\mathcal{T}^1, \ldots, \mathcal{T}^K\}$, Dyn-NAN module DN
Ensure: Updated DN over $\cup_{k=1}^{K} \mathcal{Y}^k$

for $k = 1$ to K **do**
  Sample $\tilde{\mathcal{T}}^k \subset \mathcal{T}^k$ and update $\mathcal{M} \leftarrow \mathcal{M} \cup \tilde{\mathcal{T}}^k$ ▷ **Memory update**
  if $|\mathcal{M}| > B$ **then**
    Apply random replacement
  end if
  Freeze E weights.
  for I_i^k in $\mathcal{M}$ **do** ▷ **Feature extraction**
    $x_i^k \leftarrow E(\mathrm{I}_i^k)$.
  end for
  Expand DN output layer to $\mathcal{Y}_k$ ▷ **Dynamically adapt Dyn-NAN**
  for mini-batches $\mathcal{B} \subset \mathcal{M}$ **do**
    $s_i^k \leftarrow DN(x_i^k)$. ▷ Apply Eq. 2
    Minimize cross-entropy loss $\mathcal{L}(s_i^k, y_i^k)$.
    Update DN by gradient descent
  end for
end for

- How do pre-trained models enhance the adaptability and performance of the system in RealCL scenarios?

- What is the efficiency of the novel CLARE approach in the RealCL setting and in its simplifications? How do the number of tasks and the memory budget affect the trade-off between effectiveness and efficiency for CLARE?

- What are the advantages of CLARE over state-of-the-art methods in terms of performance in continual learning scenarios? What are the underlying mechanisms that enable these advantages?

5.1. Experimental setup

Datasets: We have conducted experiments using four diverse datasets that pose distinct challenges in continual learning. The first dataset, CIFAR-10 (Krizhevsky, 2009), consists of 60,000 color images of size 32 pixels, distributed across 10 classes. Each class contains 6,000 images, and the dataset is partitioned into 50,000 training images and 10,000 test images. The second dataset, CIFAR-100 (Krizhevsky, 2009), shares similarities with CIFAR-10 but encompasses 100 classes, each containing 600 images. Similar to CIFAR-10, the dataset is divided into 50,000 training images and 10,000 test images. We also use TinyImagenet dataset (Chrabaszcz et al., 2017), a subset of the extensive ImageNet (Krizhevsky et al., 2012) collection. It features 200 classes with 500 images each, sized at 64 pixels. The dataset is further categorized into 100,000 training images, 10,000 validation images, and 10,000 test images. Finally, we also employ the CUB200 dataset (Wah et al., 2011). It consists of 11,788 images of 200 bird species. Most of the categories contain 60 images. For this dataset, we follow the same experimental setup as in previous CL work (Zhou et al., 2024).

These four are standard datasets used in the literature about continual learning. This allows us to compare our method with state-of-the-art models and demonstrate its robustness and accuracy. We here propose to use these datasets for the first time in the novel RealCL setting. For all the datasets, we use the training sets for learning the models, and report the performance using the test sets. By using four datasets with different numbers of classes, we can evaluate the performance of CLARE under various classification scenarios.

Evaluation Metrics: To evaluate the effectiveness of our approach, we have used various metrics that capture different relevant aspects of continual learning. Let $\mathcal{D} = \{\mathcal{T}^1, \mathcal{T}^2, \ldots, \mathcal{T}^K\}$ be the set of all tasks. We first employ the last task accuracy A_k, as the average accuracy of the model on the test set of the classes that it has seen until task $\mathcal{T}^k$. As a global measure of the performance of the models, we report also the average accuracy A_{Avg} over all tasks.

We quantify the method's ability to retain knowledge from previous tasks during the process of learning new tasks. This is achieved by calculating the difference in accuracy between the previous task and the current task, a measure we refer to as Global Forgetting: $F_{\text{G}} = A_{k-1} - A_k$. Then, the Average Global Forgetting is also reported, $F_{\text{AvgG}} = \frac{1}{K}\sum_{k=1}^{K}(A_{k-1} - A_k)$.

To measure the degree of forgetting in a continual learning model, we examine the reduction in accuracy on prior tasks after training on new tasks. The Task Forgetting metric is defined as,$F_T = A_k - A'_k$, where A_k is the average accuracy on the test set of classes seen up to task $\mathcal{T}^k$ before learning a new task, and A'_k is the average accuracy on the *same* test set after learning the new task $\mathcal{T}^{k+1}$. A positive value of F_T indicates forgetting, whereas a zero or negative value suggests retention or improved generalization post-learning the new task. The Average Task Forgetting is then measured by: $F_{\text{AvgT}} = \frac{1}{K}\sum_{k=1}^{K}(A_k - A'_k)$. For the first task $\mathcal{T}^1$, we assume the absence of measurable forgetting.

Implementation details: For our proposed method CLARE, we have integrated the CLIP pre-trained model (Radford et al., 2021). In the experiments, different memory sizes have been utilized for CLARE's memory module, including 2K, 4K, and 8K samples.

To populate the memory module, we follow a task-aware random sampling strategy. For each task $\mathcal{T}^k$, a subset of samples is randomly selected from the current task's data. These samples are added to the memory buffer, which also retains samples from previously seen tasks. When the buffer reaches its capacity, a replacement policy is applied that randomly discards existing samples to accommodate new ones. Importantly, this process does not enforce explicit class balancing, which aligns with the unstructured nature of the RealCL scenario. As a result, the memory buffer may contain an imbalanced representation of classes, depending on the random composition of each task.

The architecture of the Dyn-NAN module is initialized with three fully-connected layers with dimensions of 512×1024, 1024×1024, and $1024 \times$ number of classes. Two ReLU layers are positioned subsequent to the first and second fully-connected layers, serving to augment the efficacy of network operations.

To evaluate the performance of CLARE, we report the average for the evaluation metric used on the test set over five different runs for each dataset, along with the corresponding standard deviation.

We conduct all our experiments on a single machine with an Intel i7 8700 CPU, 64

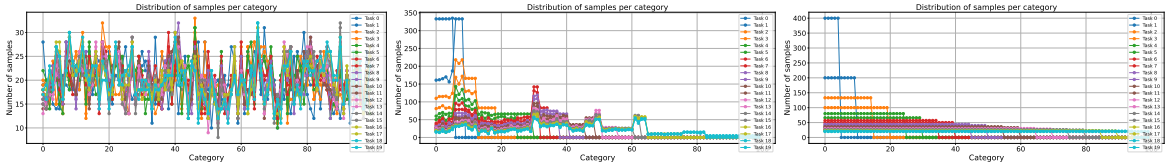
GB RAM, and a GeForce RTX 2080 Ti GPU. For the optimization, we use stochastic gradient descent (SGD) with a batch size of 64 and a weight decay of 1e-4. We also employ stochastic gradient descent with restarts (SGDR) (Loshchilov and Hutter, 2017) as a learning rate scheduler, with the following parameters: the initial cycle length $T_0 = 1$, the cycle length multiplier $T_{\text{mult}} = 2$, a warm start of one epoch, a maximum learning rate of 0.005, and a minimum learning rate of 5e-5. Moreover, we apply cutmix (Yun et al., 2019), a data augmentation technique that randomly mixes two images and their labels, with a probability of 0.5 and a regularization strength of 1.0. We apply cutmix to all datasets and RealCL scenarios, regardless of the number of classes or the size of the images.

5.2. Results

5.2.1. CLARE's Performance in all RealCL scenarios

This section starts reporting the results obtained by applying CLARE to three different continual learning scenarios proposed in Section 3: RealCL, SemiRealCL and unrealistic. We then conduct a detailed analysis and comparison of the accuracy and forgetting rates for each scenario, providing a thorough evaluation of the performance of CLARE across three of the proposed datasets with varying class numbers and memory sizes.

To better understand the implications of the proposed scenarios and the results reported in each of them, we analyze the class distributions across tasks in each setup. In the unrealistic scenario, classes are evenly distributed and introduced incrementally, resulting in balanced and non-overlapping tasks. SemiRealCL relaxes this constraint by allowing a variable number of classes per task, though without recurrence. RealCL, in contrast, introduces tasks with randomly sampled classes, allowing both imbalance and recurrence, which better reflects real-world data streams. Figure 3 illustrates these differences by showing the number of samples per class across 20 tasks when we use CLARE with the CIFAR100 dataset, for example. This visualization highlights the structural variability and unpredictability of RealCL, which poses a greater challenge for continual learning models. The lack of control over class composition and recurrence in RealCL demands adaptive mechanisms capable of handling evolving and unstructured label spaces—precisely the motivation behind our proposal.



(a) RealCL. (b) SemiRealCL. (c) Unrealistic.

Figure 3: Class distributions for all the RealCL scenarios when CLARE model is used with the CIFAR100 dataset for 20 tasks. Note that the classes have been arranged in ascending order to facilitate the visualization of the figure.

Tables 1, 2, and 3 report the results of CLARE for datasets CIFAR-10, CIFAR-100 and TinyImagenet, respectively. By analyzing all their results, we first observe that CLARE achieves the best results under the Unrealistic setup, but only in experiments conducted on the CIFAR-100 and TinyImageNet datasets. When considering the

CIFAR-10 dataset, the performance between the Unrealistic and the rest of setups is nearly equivalent. In the Unrealistic scenario, all classes are equally distributed across tasks and the memory buffer has a balanced representation of each class. Ensuring that the model has no class imbalance prevents the problem of poor performance of the model on minority classes. Interestingly, as the memory capacity increases, the model benefits from observing more samples from each class while training, which can result in higher accuracy. This proves that a greater number of examples available for generalization contribute to a more profound understanding of data by the model. However, it's important to recognize that expanding memory capacity comes with a rise in computational demands. This necessitates a considered compromise to balance the expanded memory with the overall efficiency of the model. We argue that this unrealistic scenario does not represent the best environment to validate continual learning models. The simplifications it assumes are artificial and benefit models that can leverage the potential of having well-distributed and balanced classes in memory, or that have been specifically designed for the situation where classes appear incrementally and in a controlled number. Therefore, the RealCL and SemiRealCL scenarios emerge as better candidates for evaluating the continual learning capabilities that the proposed models actually possess for this purpose.

Second, we find that in the RealCL scenario, for all datasets, CLARE reports the lowest global and task forgetting. Remember that in the RealCL scenario one does not control the sample distribution across tasks. CLARE employs a random selection process for samples stored in memory, which may result in an imbalance in the memory, characterized by varying sample counts for each class. Despite all this, CLARE is capable of offering the lowest forgetting rates, which reinforces its suitability for these more challenging scenarios. This demonstrates the effectiveness and robustness of CLARE. Furthermore, we observe that accuracy and memory capacity are positively correlated, indicating that the model advantages from a larger memory module. This shows how memory size affects CLARE performance and how using additional memory resources may lead to higher accuracy.

In the SemiRealCL setting, where the classes are randomly assigned to different tasks and have unequal distributions within each task, the memory buffer exhibits this imbalance in the way it preserves the representations of the classes. This leads to a problem where the model experiences the highest degree of forgetting, both in terms of task forgetting and global forgetting. Consequently, the last task accuracy, which reflects the model's performance after learning all the tasks, is the lowest in this scenario compared to the others, because the model is learning with a severely imbalanced memory buffer. Class imbalance in the tasks is another element that makes a difference to the average accuracy of the model because some tasks may have more classes to learn while others are simpler. However, from the tables 2 and 3, it can be seen that if the model has a larger memory budget, then its accuracy will also improve as it can hold more representations of each class and learn better.

Overall, to quantify the impact of memory size, we have conducted an in-depth analysis across all continual learning scenarios. As expected for rehearsal-based continual learning methods, increasing the replay memory consistently improves performance by preserving a more representative subset of previously observed samples. Nevertheless, the objective of these experiments is not to demonstrate that larger memories alone improve continual learning, but rather to analyze the robustness of CLARE under differ-

ent memory budgets. The observed improvements therefore reflect the complementary effect of the replay mechanism together with the proposed representation learning strategy. While it is evident that larger memory sizes improve accuracy due to enhanced class representation, our results demonstrate that they also reduce both task and global forgetting rates. This relationship is particularly crucial in realistic scenarios, where the lack of structured task organization challenges the model's ability to retain and integrate knowledge effectively. These findings underscore the importance of optimizing memory usage to balance performance and computational efficiency.

CIFAR - 10	Unrealistic	SemiRealCL	RealCL
Memory Size	M=1K	M=1K	M=1K
Last Task Accuracy	89.39 ±0.05	89.38 ±0.24	89.90 ±0.18
Average Accuracy	94.45 ±0.02	94.05 ±0.64	89.86 ±0.06
Average Global Forgetting	2.04 ±0.01	2.00 ±0.06	0.01 ±0.06
Average Task Forgetting	1.49 ±0.02	1.83 ±0.51	0.01 ±0.06

Table 1: CLARE's performance on CIFAR-10 Dataset with Memory Module size of 1K (Values in Percentage).

CIFAR - 100	Unrealistic			SemiRealCL			RealCL		
Memory Size	M=2K	M=4K	M=8K	M=2K	M=4K	M=8K	M=2K	M=4K	M=8K
Last Task Accuracy	63.95 ±0.07	69.22 ±0.06	72.98 ±0.15	56.42 ±3.24	63.11 ±2.44	68.76 ±1.80	63.11 ±0.50	68.76 ±0.11	72.26 ±0.27
Average Accuracy	77.81 ±0.01	81.07 ±0.03	83.42±0.01	79.61 ±1.68	82.35 ±1.52	84.77 ±1.41	63.19 ±0.27	68.53 ±0.10	71.88 ±0.15
Average Global Forgetting	1.79 ±0.00	1.53 ±0.00	1.34 ±0.01	2.04 ±0.17	1.71 ±0.12	1.43 ±0.10	0.00 ±0.03	-0.17 ±0.01	-0.34 ±0.01
Average Task Forgetting	1.70 ±0.04	1.42 ±0.03	1.24 ±0.02	2.28 ±0.31	1.99 ±0.23	1.74 ±0.23	0.00 ±0.03	-0.17 ±0.02	-0.34 ±0.01

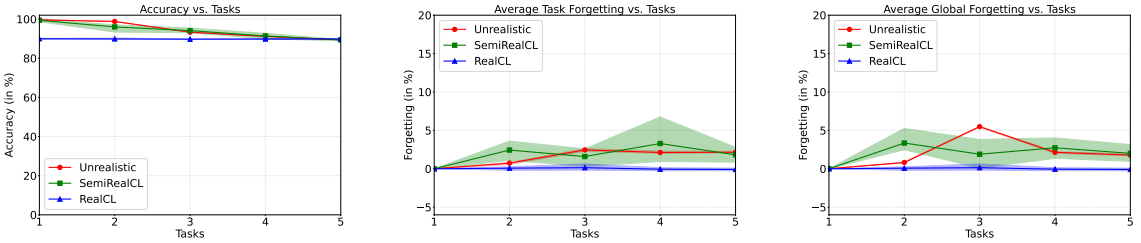
Table 2: CLARE's performance on CIFAR-100 Dataset with different Memory Module sizes (2K, 4K and 8K).

TinyImageNet	Unrealistic			SemiRealCL			RealCL		
Memory Size	M=2K	M=4K	M=8K	M=2K	M=4K	M=8K	M=2K	M=4K	M=8K
Last Task Accuracy	56.40 ±0.13	62.55 ±0.08	66.86 ±0.07	47.71 ±3.40	56.26 ±3.20	61.94 ±2.73	54.95 ±0.24	61.61 ±0.09	66.19 ±0.29
Average Accuracy	71.94 ±0.02	75.72 ±0.02	78.39 ±0.03	74.21 ±2.19	77.78 ±1.70	80.58 ±1.61	54.84 ±0.17	61.68 ±0.23	66.13 ±0.15
Average Global Forgetting	1.93 ±0.01	1.69 ±0.01	1.47 ±0.01	2.49 ±0.24	2.07 ±0.23	1.78 ±0.20	0.00 ±0.02	0.02 ±0.01	-0.11 ±0.03
Average Task Forgetting	1.97 ±0.02	1.67 ±0.04	1.45 ±0.02	2.64 ±0.41	2.24 ±0.33	1.98 ±0.31	0.00 ±0.02	0.02 ±0.01	-0.11 ±0.03

Table 3: CLARE's performance on TinyImageNet Dataset with different Memory Module sizes (2K, 4K and 8K).

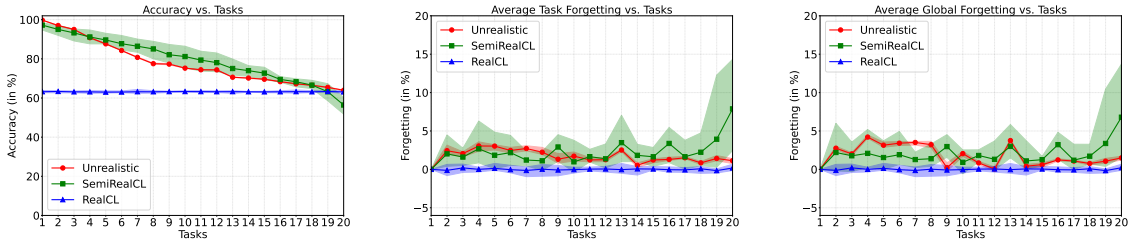
Figures 4 to 10 show the detailed results for different scenarios and datasets. CLARE model reports the highest accuracy on the CIFAR10 dataset, which has only 10 classes, and can classify and learn them in all scenarios. However, CIFAR100 and TinyImagenet datasets have more classes (100 and 200, respectively), which makes the classification task more challenging. The average accuracies of CLARE model on these datasets are lower, but still comparable to other state-of-the-art methods. Moreover, it is noteworthy that CLARE model achieves this accuracy by only training on the existing samples in the memory buffer. Interestingly, these results indicate that forgetting is indeed mitigated in the realistic setting. A detailed analysis reveals the following aspects and causes. First, we believe that the main reason is due to the artificiality of the scenario: it is not a purely realistic scenario, but rather a simulation of a real situation, with

a closed data set, such as CIFAR-100 or TinyImageNet. This makes the difficulty of the scenario less than in a real situation, where not handling a closed set of classes would present a greater variability over the training time of the models. Nevertheless, our RealCL scenario is much more realistic than previous proposals handled in the literature. Second, forgetting has been mitigated, as our experiments reveal, by our own CLARE model. The minimal performance degradation observed in the realistic setting can be attributed to CLARE's architecture, which leverages the fixed representations of a robust pre-trained model alongside the dynamic adjustment capabilities of Dyn-NAN. This design effectively mitigates forgetting by isolating task-specific adaptations within the Dyn-NAN module while preserving shared knowledge in the pre-trained encoder. While forgetting is not entirely absent, the reduced degradation highlights CLARE's capacity to handle the unstructured nature of the realistic incremental setup, as demonstrated by its low global and task forgetting rates. These results emphasize the model's ability to balance stability and plasticity effectively.



(a) Task accuracy. (b) Average task forgetting. (c) Average global forgetting.

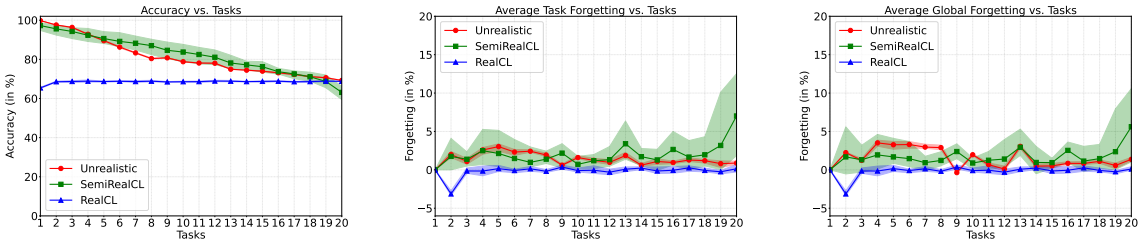
Figure 4: Results of CLARE for CIFAR-10 in the three scenarios with memory module size of 1K.



(a) Task accuracy. (b) Average task forgetting. (c) Average global forgetting.

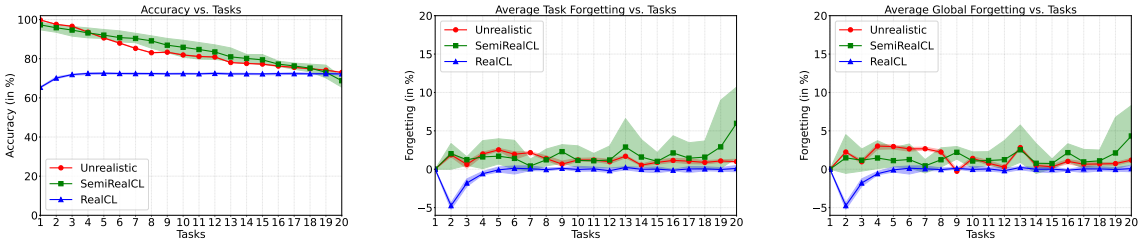
Figure 5: Results of CLARE for CIFAR-100 in the three scenarios with memory module size of 2K.

As the size of the memory module increases, CLARE demonstrates improved performance metrics. The question that we want to address now is: How does CLARE perform when the number of tasks increases? We have evaluated CLARE, which uses a powerful pre-trained model, on the RealCL and SemiRealCL settings. These scenarios pose more challenges as the number of tasks increases. Classic continual learning models are prone to catastrophic forgetting, which means losing old knowledge when learning new tasks. This aspect can be relevant specially when the number of tasks grows. We have performed the evaluation using the CIFAR-100 dataset with 2K memory size. Table 4 shows that CLARE can keep its performance on all tasks, even in the RealCL setup. Additionally, in the SemiRealCL scenario CLARE outperforms the results obtained for 20 tasks in terms of Last Task Accuracy (see Figure 11).



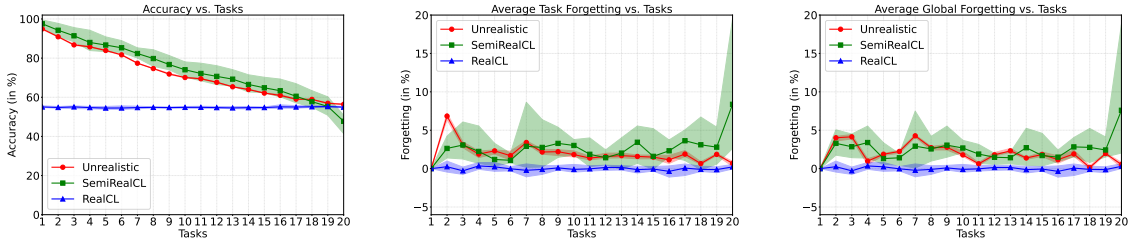
(a) Task accuracy. (b) Average task forgetting. (c) Average global forgetting.

Figure 6: Results of CLARE for CIFAR-100 in the three scenarios with memory module size of 4K.



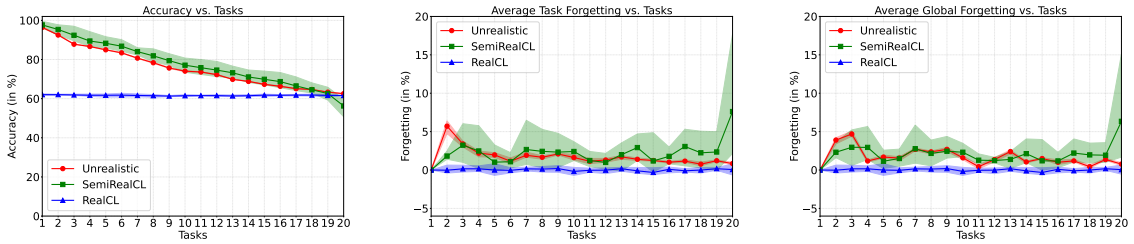
(a) Task accuracy. (b) Average task forgetting. (c) Average global forgetting.

Figure 7: Results of CLARE for CIFAR-100 in the three scenarios with memory module size of 8K.



(a) Task accuracy. (b) Average task forgetting. (c) Average global forgetting.

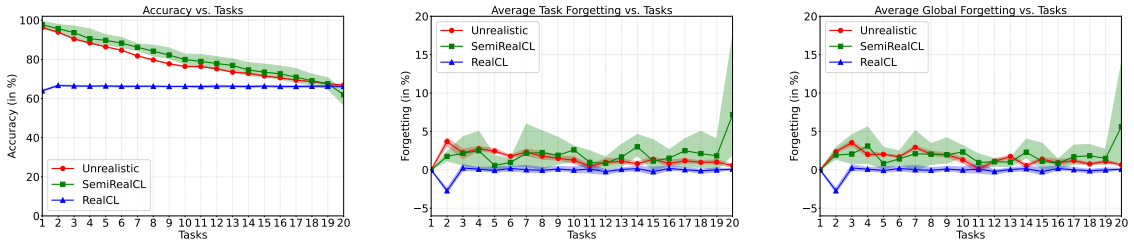
Figure 8: Results of CLARE for TinyImagenet in the three scenarios with memory module size of 2K.



(a) Task accuracy. (b) Average task forgetting. (c) Average global forgetting.

Figure 9: Results of CLARE for TinyImagenet in the three scenarios with memory module size of 4K.

These results reinforce the observation that, although increasing the replay memory generally leads to better performance, the effectiveness of CLARE cannot be explained solely by the replay buffer. Notably, when the number of incremental tasks increases from 20 to 50 in the CIFAR-100 benchmark, CLARE maintains consistently strong performance using the same 2K memory budget despite the substantially higher task



(a) Task accuracy. (b) Average task forgetting. (c) Average global forgetting.

Figure 10: Results of CLARE for TinyImagenet in the three scenarios with memory module size of 8K.

fragmentation. This suggests that the proposed learning strategy effectively complements the replay mechanism, enabling the model to preserve knowledge under increasingly challenging continual learning conditions. Therefore, the replay buffer should be regarded as an enabling component of CLARE rather than as the sole explanation for its performance.

CIFAR - 100	SemiRealCL	RealCL
Number of Tasks	50 Tasks	50 Tasks
Last Task Accuracy	61.23 ±0.71	63.08 ±0.06
Average Accuracy	77.72 ±2.40	63.01 ±0.52
Average Global Forgetting	0.74 ±0.00	-0.01 ±0.00
Average Task Forgetting	0.72 ±0.03	0.32 ±0.15

Table 4: Performance metrics for 50 Tasks on the CIFAR-100 dataset with a 2K memory module in SemiRealCL and RealCL scenarios (Values in Percentage).

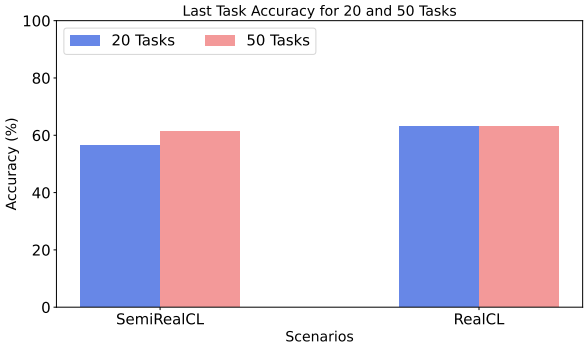


Figure 11: Comparative analysis of the last task accuracy of CLARE for 20 and 50 tasks in both SemiRealCL and RealCL scenarios.

5.2.2. Comparison with state-of-the-art models

We report a comparison of CLARE with two recent and representative continual learning methods: MEMO (Zhou et al., 2023) and APER (Zhou et al., 2024). They are strong state-of-the-art baselines that exemplify two relevant paradigms, memory-based rehearsal and pre-trained model adaptation, respectively. At the time of submisision,

both MEMO and APER reported the best results in traditional continual learning settings, i.e. our unrealistic scenario. Note that the methods to be compared with CLARE do not all employ the same backbone architectures. Consequently, the reported results should not be interpreted as an isolated comparison of feature extractors, but as a comparison of performance when we move from traditional unrealistic scenarios to the proposed RealCL one, with state-of-the-art models. For this reason, we compare complete continual learning systems following the experimental protocols adopted in the corresponding original works.

MEMO (Zhou et al., 2023) is a sophisticated method that integrates a memory module to preserve and recall significant instances from prior classes in a continual learning setting. MEMO's technique creatively introduces extra residual layers instead of reconstructing the full backbone, effectively reducing memory expenses. MEMO has been specifically developed for class-incremental scenarios. In fact, it includes a sample selection mechanism for the memory buffer it employs, which requires categories to sequentially arrive. To tailor MEMO for our new RealCL paradigm, we have had to adapt this selection mechanism. In RealCL, training samples are randomly allocated across different tasks without predetermined class distributions within each task. This allows for the possibility of class recurrence across different tasks. Given this setup, a task may not provide access to all samples from a class, as MEMO needs, thereby limiting the selection of the most representative samples available within that task. Consequently, there is no predetermined quota for the number of samples per class in the memory buffer. We initially select the most significant samples from the current task's classes using MEMO's sample selection method based on (Welling, 2009). These chosen samples are then added to the memory buffer through a random selection process that balances between the important samples of previous tasks and the new samples of the current task. To accommodate the new samples, we clear out an equivalent space in the memory buffer, ensuring that the space is equally distributed among the classes presented in each task. These samples, once chosen, are incorporated into the training data for subsequent tasks. MEMO model (Zhou et al., 2023), which utilizes a ResNet32-based architecture (He et al., 2015), processes the data from the current task alongside the representative samples from previous tasks preserved in the memory buffer. Consequently, the absence of new classes may not necessitate significant alterations or expansions in the specialized layers, as their primary role is to adapt to the nuances of new class features.

APER (Zhou et al., 2024) belongs to the family of pre-trained models for continual learning, like our CLARE approach. APER employs a pre-trained model as input for an elaborated adaptation process that integrates some parameter efficient tuning techniques. As it is reported by Zhou et al. (2024), APER obtains the best results when it is combined with the Adaptformer (Chen et al., 2022) technique. Therefore, this is the version of APER that we have decided to integrate in our novel RealCL paradigm. As for the implementation of APER, we have used the official one released with (Sun et al., 2023).

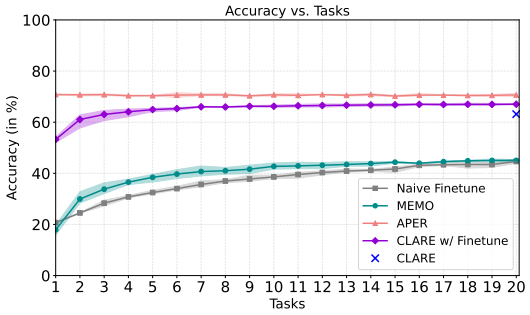
Finally, we also include in this analysis a baseline model, termed naive fine-tuning, that simply involves a learning strategy that consists in updating the model's parameters with newly acquired data, albeit in the absence of any regularization mechanisms or memory-based strategies. The fine-tuning is conducted using a ResNet-32 architecture (He et al., 2015). Given the RealCL scenario, where classes may reappear across

various tasks, the model is less likely to experience significant forgetting. It has the potential to continually refine the representations of previous classes as new instances of these classes emerge in subsequent tasks.

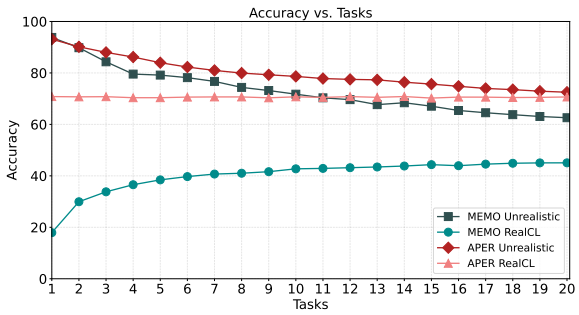
We start with the comparison with the state of the art in the CIFAR-100 dataset. Figure 12a and Table 5 show the results obtained in our experiments. Interestingly, both MEMO and APER experience a reduction in performance when adapted and evaluated in the novel RealCL scenario. As it was reported in (Zhou et al., 2023), in the standard unrealistic scenario, MEMO achieved a Last Task Accuracy of 62.59% after 20 tasks, whereas in our RealCL implementation, it reports 45.06%. For the APER model, the Last Task Accuracy in the unrealistic setting was of 72.5%, while for our RealCL we report 70.66%. We argue that MEMO, APER and their traditional competitors perform well in an ideal CL setting (i.e. our unrealistic scenario) with complete access to class distributions and specific data samples, where the setting follows a pure class-incremental process. However, their effectiveness tend to decrease in the RealCL scenario. It is relevant to compare, in fact, how these algorithms evolve in both scenarios, the RealCL and the unrealistic or traditional one. In Figure 12b we show this interesting comparison. Clearly, our realistic scenario pushes these models to their limits, which is reflected in a drop in their accuracy and even in a learning stagnation, as observed in the APER model, whose accuracy remains stuck from the first task.

CIFAR - 100	Naive Fine-tuning	MEMO (Zhou et al., 2023)	APER (Zhou et al., 2024)	CLARE	CLARE with Fine-tuning
Number of Tasks	20 Tasks	20 Tasks	20 Tasks	20 Tasks	20 Tasks
Last Task Accuracy	44.63 ±0.73	45.06 ±0.37	70.66 ±0.51	63.39 ±0.40	67.03 ±0.22
Average Accuracy	37.07 ±6.73	40.17 ±6.60	70.58 ±0.25	63.34 ±0.14	65.20 ±3.20
Average Task Forgetting	-1.20 ±1.10	-1.35 ±2.70	-0.17 ±0.25	-0.002 ±0.22	-0.68 ±1.74

Table 5: Comparison with state-of-the-art models in the RealCL scenario. Performance Metrics on CIFAR-100 Dataset with a 2K Buffer Size for CLARE. Values in Percentage.



(a) Comparison with state-of-the-art models in the RealCL scenario.



(b) Comparison of MEMO and APER models in the RealCL and the Unrealistic scenarios.

Figure 12: Results in CIFAR-100 dataset. (a) Performance comparison of CLARE with and without fine-tuning against various state-of-the-art models in RealCL scenario. (b) Comparative analysis of the performance of MEMO and APER in unrealistic scenario versus RealCL outcomes.

The results for the experimental evaluation with the dataset CUB200 are shown in Table 6. Here, CLARE, with and without fine-tuning, improves the performance of APER according to all metrics.

Our main conclusions of this comparison conducted with state-of-the-art methods are as follows. First, methods originally proposed for traditional continual learning

CUB - 200	APER (Zhou et al., 2024)	CLARE	CLARE with Fine-tuning
Number of Tasks	20 Tasks	20 Tasks	20 Tasks
Last Task Accuracy	55.73 ±0.025	61.41 ±0.66	66.48 ±0.91
Average Accuracy	55.42 ±1.34	59.36 ±0.10	62.42 ±0.87
Average Task Forgetting	-0.04 ±1.81	-0.67 ±0.09	-0.92 ±0.09

Table 6: Performance metrics for 20 Tasks on the CUB200 dataset, with a 2K memory module, in a RealCL scenario. We compare APER and CLARE (with and without fine-tunning). Values in Percentage.

benchmarks experience a substantial performance degradation under the novel RealCL setting, suggesting that state-of-the-art performance in conventional unrealistic benchmarks does not necessarily translate to more realistic continual learning scenarios. With our study we show that it is important to subject CL models to a scenario like RealCL, where possible deficiencies are not masked when class distributions do not appear artificially and incrementally ordered with tasks. Interestingly, our observations revealed that the naive fine-tuning model, APER and MEMO demonstrate negative values in task forgetting metrics. This suggests that the models are not merely retaining previously acquired knowledge, but are also contributing to an enhancement in overall accuracy as they are exposed to new data from subsequent tasks. However, this trend appears to plateau once the models have processed a certain number of tasks, indicating a stabilization point beyond which the introduction of additional data does not yield significant improvements in model performance. This effect is particularly relevant for APER, which does not seem to continue learning after the first task (see Figure 12b). This aligns with how APER is trained, as detailed in (Zhou et al., 2024). APER adapts the pre-trained backbone only in the first task and then employs a strategy based on prototypes for every category, along with parameter fine-tuning mechanisms such as Adaptformer (Chen et al., 2022). While this strategy seems effective for an the unrealistic scenario, i.e. the standard class-incremental learning scenario, it is not suitable for our new RealCL setup.

Second, our novel CLARE model presents a compelling and competitive option for realistic scenarios. In the experiments with CIFAR-100, the APER model reports a last task accuracy slightly higher than CLARE. However, this is not the case with the CUB-200 dataset, where CLARE achieves better results in the realistic scenario. It is worth mentioning that APER, as detailed by Zhou et al. (2024), is a significantly more complex model than CLARE, integrating multiple parameter-efficient tuning techniques along with the pre-trained model. These results indicate that the observed improvements cannot be attributed solely to the use of a stronger visual backbone, but also to the proposed learning strategy within the RealCL framework. CLARE depends on a robust pre-trained model capable of identifying general and adaptable features from data. It is complemented by a learning-based memory module that actively selects and updates a set of samples for each task. This dual strategy enables our method to maintain a balance between preserving existing knowledge and assimilating new information, thereby circumventing the pitfalls of catastrophic forgetting or interference. As demonstrated in Tables 5 and 6, adopting a fine-tuning approach with CLARE leads to improved accuracy across tasks. It is important to note that since the pre-trained model within CLARE remains static, fine-tuning is applied solely to adjust the learned weights of the Dyn-NAN block. Further observations reveal that CLARE, upon fine-

tuning, also displays negative task forgetting, indicating an expansion in the model's learning capabilities as the number of tasks increases.

6. Conclusions

With this paper, we had two main objectives. The first one was to propose to the community the RealCL paradigm for evaluating CL solutions. The novel RealCL formulation has allowed us to explore up to three distinct continual learning scenarios, ranging from a highly structured scenario with controlled data and class distribution to a completely uncontrolled and realistic setting. We conducted a thorough evaluation of our proposed scenarios employing CLARE, a straightforward and model-agnostic methodology that utilizes pre-trained models to facilitate the learning process. This was the second objective of our work. We believe that especially in these unstructured scenarios, solutions based on pre-trained models can play a fundamental role. Our CLARE model enables the seamless integration of new knowledge while preserving previously acquired information, eliminating the need for extensive retraining. Moreover, we implemented the proposed realistic scenario for a set of state-of-the-art models and a baseline, and assessed their performance, comparing it with the outcomes achieved by CLARE. Our findings highlight the versatility and robustness of CLARE, particularly when contending with the unpredictability of real-world data. The code to reproduce all our results can be found at https://github.com/gramuah/clare.

Limitations and future work:

While CLARE demonstrates strong performance in realistic continual learning scenarios, certain limitations should be acknowledged. First, CLARE relies on a compact replay memory, which may introduce privacy concerns or storage constraints in some applications. Although the memory is intentionally bounded and contains only a small set of representative samples, the method still depends on storing part of the past experience. In practice, this design represents a compromise between purely memory-free approaches, which often suffer from severe forgetting, and unrestricted rehearsal strategies that require access to large historical datasets. CLARE mitigates forgetting effectively by combining a small replay buffer with a frozen pre-trained model, but its performance remains influenced by the memory size and the sample selection strategy, both of which could be further optimized. Additionally, the use of a frozen pre-trained model provides stability and strong transfer capabilities, but it may limit adaptability in highly specialized or rapidly evolving domains. Investigating adaptive fine-tuning strategies that preserve stability while improving domain adaptation constitutes an important direction for future work. Furthermore, although CLARE has been evaluated on standard continual learning benchmarks, its behavior under stronger domain shifts and highly imbalanced real-world data distributions deserves further investigation.

Regarding our second contribution, the proposed RealCL scenario, it is important to emphasize that RealCL defines a continual learning protocol rather than a specific algorithm. One of the assumptions of RealCL is that the complete historical datasets from previous tasks are no longer available once learning progresses. However, this formulation is compatible with both replay-based and replay-free continual learning methods. Therefore, the use of a bounded replay buffer in CLARE, or in any other replay-based model, should be understood as an algorithmic choice rather than a requirement imposed by RealCL. In privacy-sensitive applications, replay-free approaches

may be preferable, and evaluating such methods under the RealCL protocol constitutes an interesting avenue for future work. Finally, although RealCL is designed to be closer to real-world continual learning conditions, it still assumes that data arrive in discrete tasks rather than as a fully continuous stream. Future research could explore continual learning systems that dynamically integrate incoming data without relying on predefined task boundaries, further narrowing the gap between benchmark protocols and a real-world deployment scenario.

Acknowledgements

This research was partially funded by projects: GUIDANCE-4D, with reference CM/DEMG/2024-028, of CAM-UAH; NAVIGATOR-D, with reference PID2023-148310OB-I00 from the Ministry of Science and Innovation of Spain; NAVISOCIAL, with reference 2023/00405/001 from the University of Alcalá; and MOEVE Chair at the University of Alcalá.

References

Aljundi, R., Lin, M., Goujaud, B., Bengio, Y., 2019. Gradient based sample selection for online continual learning, in: Neural Information Processing Systems.

Araujo, V., Hurtado, J., Soto, Á., Moens, M.F., 2022. Entropy-based stability-plasticity for lifelong learning. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) , 3720–3727.

Buzzega, P., Boschini, M., Porrello, A., Calderara, S., 2020. Rethinking experience replay: a bag of tricks for continual learning. 2020 25th International Conference on Pattern Recognition (ICPR) , 2180–2187.

Cao, B., Tang, Q., Lin, H., Han, X., Chen, J., Wang, T., Sun, L., 2024. Retentive or forgetful? diving into the knowledge memorizing mechanism of language models, in: LREC-COLING.

Castro, F.M., Marín-Jiménez, M.J., Mata, N.G., Schmid, C., Karteek, A., 2018. End-to-end incremental learning, in: European Conference on Computer Vision.

Chen, C., Min, W., Li, X., Jiang, S., 2019. Hybrid incremental learning of new data and new classes for hand-held object recognition. Journal of Visual Communication and Image Representation 58, 138–148.

Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P., 2022. Adaptformer: Adapting vision transformers for scalable visual recognition, in: NeurIPS.

Chrabaszcz, P., Loshchilov, I., Hutter, F., 2017. A downsampled variant of imagenet as an alternative to the cifar datasets. ArXiv abs/1707.08819.

De Lange, M., Tuytelaars, T., 2021. Continual prototype evolution: Learning online from non-stationary data streams, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 8230–8239.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale, in: ICLR.

Douillard, A., Cord, M., Ollion, C., Robert, T., Valle, E., 2020. Podnet: Pooled outputs distillation for small-tasks incremental learning, in: European Conference on Computer Vision.

Douillard, A., Ram'e, A., Couairon, G., Cord, M., 2021. Dytox: Transformers for continual learning with dynamic token expansion. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 9275–9285.

French, R.M., 1993. Catastrophic interference in connectionist networks: Can it be predicted, can it be prevented?, in: Neural Information Processing Systems.

Goodfellow, I.J., Mirza, M., Da, X., Courville, A.C., Bengio, Y., 2014. An empirical investigation of catastrophic forgeting in gradient-based neural networks, in: 2nd International Conference on Learning Representations, ICLR 2014.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 770–778.

Hu, J.E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Chen, W., 2021. Lora: Low-rank adaptation of large language models. ArXiv abs/2106.09685.

Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S.J., Hariharan, B., Lim, S.N., 2022. Visual prompt tuning. ArXiv abs/2203.12119.

Jung, D., Han, D., Bang, J., Song, H., 2023. Generating instance-level prompts for rehearsal-free continual learning. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) , 11813–11823.

Krizhevsky, A., 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Communications of the ACM 60, 84 – 90.

Lee, A., Gomes, H.M., Zhang, Y., 2022. Balancing the stability-plasticity dilemma with online stability tuning for continual learning. 2022 International Joint Conference on Neural Networks (IJCNN) , 1–8.

Li, Z., Hoiem, D., 2016. Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence 40, 2935–2947.

Lomonaco, V., Pellegrini, L., López, P.R., Caccia, M., She, Q., Chen, Y., Jodelet, Q., Wang, R., Mai, Z., Vázquez, D., Parisi, G.I., Churamani, N., Pickett, M., Laradji, I.H., Maltoni, D., 2020. Cvpr 2020 continual learning in computer vision competition: Approaches, results, current challenges and future directions. Artif. Intell. 303, 103635.

Lopez-Paz, D., Ranzato, M., 2017. Gradient episodic memory for continual learning, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, p. 6470–6479.

Loshchilov, I., Hutter, F., 2017. SGDR: Stochastic gradient descent with warm restarts, in: ICLR.

McCloskey, M., Cohen, N.J., 1989. Catastrophic interference in connectionist networks: The sequential learning problem. Psychology of Learning and Motivation 24, 109–165.

Mermillod, M., Bugaiska, A., Bonin, P., 2013. The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. Frontiers in Psychology 4.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I., 2021. Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning.

Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H., 2017. icarl: Incremental classifier and representation learning, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5533–5542.

Shi, L., Zhao, K., Fu, Z., 2023. Boosting separated softmax with discrimination for class incremental learning. Journal of Visual Communication and Image Representation 95, 103899.

Shim, D., Mai, Z., Jeong, J., Sanner, S., Kim, H.J., Jang, J., 2020. Online class-incremental continual learning with adversarial shapley value, in: AAAI Conference on Artificial Intelligence.

Shmelkov, K., Schmid, C., Alahari, K., 2017. Incremental learning of object detectors without catastrophic forgetting, in: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 3420–3429.

Smith, J., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R.S., Kira, Z., 2022. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 11909–11919.

Sun, H.L., Zhou, D.W., Ye, H.J., Zhan, D.C., 2023. Pilot: A pre-trained model-based continual learning toolbox. arXiv preprint arXiv:2309.07117 .

Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: Neural Information Processing Systems.

van de Ven, G.M., Tolias, A.S., 2019. Three scenarios for continual learning. ArXiv abs/1904.07734.

van de Ven, G.M., Tuytelaars, T., Tolias, A.S., 2022. Three types of incremental learning. Nature Machine Intelligence 4, 1185 – 1197.

Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S., 2011. The Caltech-UCSD birds-200-2011 dataset .

Wang, F.L., Zhou, D.W., Liu, L., Ye, H.J., Bian, Y., chuan Zhan, D., Zhao, P., 2023. Beef: Bi-compatible class-incremental learning via energy-based expansion and fusion, in: International Conference on Learning Representations.

Wang, Y., Huang, Z., Hong, X., 2022a. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning, in: NeurIPS.

Wang, Y., Ma, Z., Huang, Z., Wang, Y., Su, Z., Hong, X., 2022b. Isolation and impartial aggregation: A paradigm of incremental learning without interference, in: AAAI Conference on Artificial Intelligence.

Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.Y., Ren, X., Su, G., Perot, V., Dy, J.G., Pfister, T., 2022c. DualPrompt: Complementary prompting for rehearsal-free continual learning, in: ECCV.

Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J.G., Pfister, T., 2021. Learning to prompt for continual learning. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 139–149.

Welling, M., 2009. Herding dynamical weights to learn, in: Proceedings of the 26th Annual International Conference on Machine Learning, Association for Computing Machinery, New York, NY, USA. p. 1121–1128.

Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.J., 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) , 6022–6031.

Zhao, B., Xiao, X., Gan, G., Zhang, B., Xia, S., 2019. Maintaining discrimination and fairness in class incremental learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 13205–13214.

Zhao, H., Wang, H., Fu, Y., Wu, F., Li, X., 2020. Memory-efficient class-incremental learning for image classification. IEEE Transactions on Neural Networks and Learning Systems 33, 5966–5977.

Zhou, D.W., Cai, Z.W., Ye, H.J., Zhan, D.C., Liu, Z., 2024. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. International Journal of Computer Vision doi:10.1007/s11263-024-02218-0.

Zhou, D.W., Wang, Q., Ye, H.J., Zhan, D.C., 2023. A model or 603 exemplars: Towards memory-efficient class-incremental learning, in: ICLR.

Zhou, K., Yang, J., Loy, C.C., Liu, Z., 2021. Learning to prompt for vision-language models. International Journal of Computer Vision 130, 2337 – 2348.

Zhou, P., Mai, L., Zhang, J., Xu, N., Wu, Z., Davis, L.S., 2019. M2kd: Multi-model and multi-level knowledge distillation for incremental learning. ArXiv abs/1904.01769.