Machine Learning
https://doi.org/10.1007/s10994-024-06524-z

From MNIST to ImageNet and back: benchmarking continual curriculum learning

Kamil Faber[1] · Dominik Zurek[1] · Marcin Pietron[1] · Nathalie Japkowicz[3] · Antonio Vergari[2] · Roberto Corizzo[3]

Received: 6 March 2023 / Revised: 4 November 2023 / Accepted: 14 February 2024

Abstract
Continual learning (CL) is one of the most promising trends in recent machine learning research. Its goal is to go beyond classical assumptions in machine learning and develop models and learning strategies that present high robustness in dynamic environments. This goal is realized by designing strategies that simultaneously foster the incorporation of new knowledge while avoiding forgetting past knowledge. The landscape of CL research is fragmented into several learning evaluation protocols, comprising different learning tasks, datasets, and evaluation metrics. Additionally, the benchmarks adopted so far are still distant from the complexity of real-world scenarios, and are usually tailored to highlight capabilities specific to certain strategies. In such a landscape, it is hard to clearly and objectively assess models and strategies. In this work, we fill this gap for CL on image data by introducing two novel CL benchmarks that involve multiple heterogeneous tasks from six image datasets, with varying levels of complexity and quality. Our aim is to fairly evaluate current state-of-the-art CL strategies on a common ground that is closer to complex real-world scenarios. We additionally structure our benchmarks so that tasks are presented in increasing and decreasing order of complexity—according to a curriculum—in order to evaluate if current CL models are able to exploit structure across tasks. We devote particular emphasis to providing the CL community with a rigorous and reproducible evaluation protocol for measuring the ability of a model to generalize and not to forget while learning. Furthermore, we provide an extensive experimental evaluation showing that popular CL strategies, when challenged with our proposed benchmarks, yield sub-par performance, high levels of forgetting, and present a limited ability to effectively leverage curriculum task ordering. We believe that these results highlight the need for rigorous comparisons in future CL works as well as pave the way to design new CL strategies that are able to deal with more complex scenarios.

Keywords Continual learning · Lifelong learning · Curriculum learning · Neural networks · Computer vision · Image classification

Editors: Dino Ienco, Roberto Interdonato, Pascal Poncelet.

Antonio Vergari, Roberto Corizzo: Equal Supervision.

Extended author information available on the last page of the article

Published online: 22 April 2024

1 Introduction

Continual Learning (CL), also known as Lifelong Learning, is a promising learning paradigm to design models that have to learn how to perform *multiple tasks* across different environments over their lifetime (Parisi et al., 2019).[1] Ideal CL models in the real world should be able to quickly adapt to new environments and tasks while perfectly retaining what they learned in the past, thus only increasing, and not decreasing, their performance as they experience more tasks. In practice, this is quite challenging due to the hardness of generalizing from one environment to another when there is a huge *distribution shift* between them (Cano & Krawczyk, 2022; David Lopez-Paz, 2017; Krawczyk, 2021; Li & Hoiem, 2017), and to the fact that models tend to (sometimes catastrophically) *forget* what they learned for previous tasks.

The great attention around this paradigm has brought many communities to focus on how to address these challenges, including reinforcement learning (Abel et al., 2018; Baker et al., 2023) and anomaly detection (Corizzo et al., 2022; Faber et al., 2022b). It is noteworthy that significant efforts in CL have been devoted to computer vision, leading to a large number of proposed models (Aljundi et al., 2018a; Chaudhry et al., 2019; David Lopez-Paz, 2017; Hihn & Braun, 2022; Kang et al., 2022; Li & Hoiem, 2017; Rolnick et al., 2019; Zenke et al., 2017), where the most common task is to learn models that can classify different kinds of images while preventing catastrophic forgetting or quickly adapting to new image classes or image datasets. Every new model has been evaluated in a slightly different setting – *using a different dataset, evaluation metrics and learning protocols*—thus generating a number of CL learning and evaluation schemes. The result is that *the benchmark panorama of CL in computer vision is quite fragmented*, and therefore it has become tougher to measure catastrophic forgetting and domain adaptation in a fair and homogeneous way for the many CL models we have in the literature these days. Furthermore, all previous evaluation protocols are designed to highlight some specific model characteristics and, as such, are generally over-simplified w.r.t. real-world data (Cossu et al., 2022).

For example, one of the most popular evaluation protocols for CL models in computer vision is to design different tasks to classify different (subsets of the) classes of a single dataset (Lange et al., 2022; Van de Ven & Tolias, 2019). The most prominent example is splitMNIST in which the 10 digits from MNIST (Cun) are (usually) divided into 5 tasks consisting of 2 digits each. Similar approaches are proposed for CIFAR10 (Krizhevsky, 2009), and TinyImagenet (Le & Yang, 1998). Other datasets, such as Continuous Object Recognition (CORe50) (Lomonaco & Maltoni, 2017), specifically designed for CL, still make the same assumptions to generate tasks. Clearly, these protocols are not suited to detect distribution shifts due to the high inter-task similarity. Consequently, catastrophic forgetting is much easier to prevent in these cases. Therefore the reported metrics for models evaluated in this way can be overly optimistic.

To deal with domain shifts, researchers have recently started to sample tasks from two different datasets. For instance, David Lopez-Paz (2017) proposed to train and evaluate a model on Imagenet first and then challenge its performance on the Places365 dataset. Li and Hoiem (2017) considers more scenarios, starting with Imagenet or Places365, and then moving on to the VOC/CUB/Scenes datasets. Few works propose more advanced scenarios built on top of more than two datasets. The two most prominent examples are the so-called

[1] To uniform the language and enhance the readability of the paper we adopt the unique term continual learning (CL).

Table 1 Benchmarks comparison considering only multi-dataset benchmarks

Benchmark	i)	ii)	iii)	iv)	v)	CI)	TI)
Imagenet/Places365 to VOC/CUB/Scenes [35]	✓	x	x	✓	✓	x	✓
Imagenet to Places365 [41]	✓	x	x	✓	✓	x	✓
5-Datasets [20]	✓	✓	x	✓	✓	✓	x
RecogSeq [3] [33]	✓	✓	x	✓	✓	✓	✓
M2I, I2M *(ours)*	✓	✓	✓	✓	✓	✓	✓

Columns refer to: (i) supporting multiple heterogeneous tasks; (ii) varying task complexity and quality; (iii) evaluating curriculum strategies; (iv) rigorous way to measure generalization and forgetting; and (v) exactly reproducible out-of-the-box. In addition, we consider the coverage of class (CI) and task-incremental (TI) learning settings. The symbols have the following meaning:✓—criterion is covered; ✓—criterion is covered at some part or with some limitations; x—criterion is not covered at all

5-datasets (Ebrahimi et al., 2020) and RecogSeq (Aljundi et al., 2018b), which provide models with more challenging scenarios than previous attempts, increasing the number of considered datasets to 5 and 8, respectively. Unfortunately, those datasets provide a similar task complexity due to the limited differences across datasets. Furthermore, when different datasets are employed, it is important to "calibrate the meaning" of the employed metrics, taking into account the number of classes involved in each task.

Despite all this progress, we argue that there is still not a robust and standardized evaluation benchmark for the many CL models in the literature. We argue that a modern benchmark for CL should provide the following aspects. First, ***multiple heterogeneous tasks*** that do not restrict to a single set of concepts, e.g., digits in MNIST or SVHN or naturalistic images as in Imagenet or CIFAR10. Second, ***a varying quality and complexity of the tasks***, e.g., alternating from black and white (B &W) to RGB images and vice-versa, considering different image sizes, and a number of concepts. Third, a way to systematically evaluate if ***learning on a curriculum*** of task complexities help with domain generalization and catastrophic forgetting. For example, evaluating if a model trained on B &W digits can better generalize to B &W letters and then to RGB digits and letters, or if learning them in the inverse order is more beneficial. Fourth, ***a rigorous way to measure generalization and forgetting*** in terms of modern backward and forward transfer metrics Díaz-Rodríguez et al. (2018) in a number of different evaluation scenarios, i.e., when classes or tasks are introduced incrementally (Van de Ven & Tolias, 2019). Lastly, all results should be ***exactly reproducible out-of-the-box***. We argue that all the previous CL works discussed above do not consider one or more of these criteria, as highlighted in Table 1. In addition to the five desiderata, we also cover both class and task-incremental learning settings, which is not usually the case for other surveyed works. In this paper, we aim to overcome these limitations.

Specifically, the contributions of the paper are as follows:

- We propose *a set of benchmarks* built on 6 image datasets ordered in a curriculum of complexity—*from MNIST to TinyImageNet* (M2I) and back *from TinyImageNet to MNIST* (I2M)—that simultaneously satisfies all the above desiderata. These benchmarks have varying task complexity, starting with simple digits and going to complex naturalistic images and vice versa (see Fig. 1);
- We provide an exhaustive experimental evaluation including 10 state-of-the-art continual learning methods, covering the key categories of approach (architectural, regularization, and rehearsal) in both class and task-incremental settings, which natu-

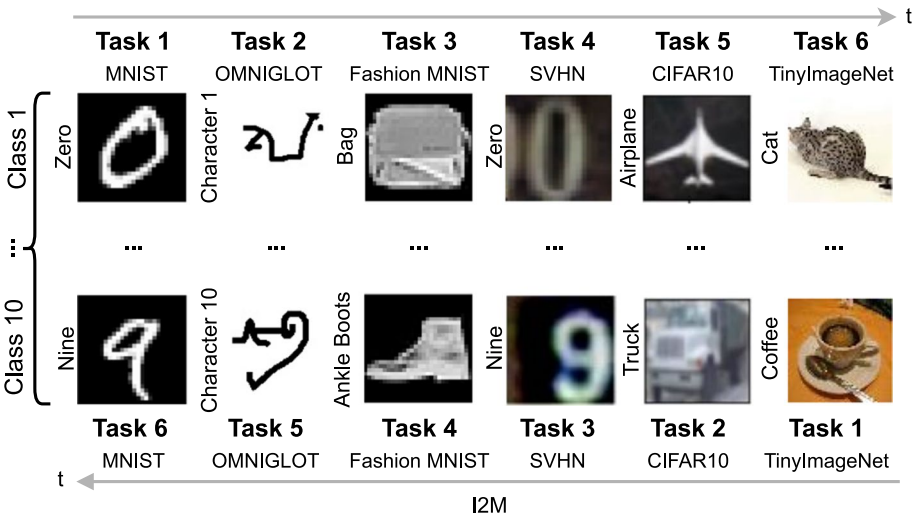

Fig. 1 The Proposed M2I and I2M continual learning benchmarks. There are 6 different tasks, each sampled from a different dataset: MNIST (Cun), OMNIGLOT (Lake et al., 2015), Fashion MNIST (Xiao et al., 2017), SVHN (Netzer et al., 2011), CIFAR10 (Krizhevsky, 2009) and TinyImageNet (Le & Yang, 1998). Tasks are organized in two curriculum ordering, from simple to harder (left to right) and backward (right to left). Every task sports 10 classes, as to make the performance metric meaning intuitive and faithful

rally fit our multi-task curriculum learning scenario, and evaluating results using the most recent metrics adopted in the continual and lifelong learning community.

2 Background

2.1 CL scenario types

A wide range of scenarios was designed and discussed in recent studies to design effective CL models while trying to reflect real-world challenges. In image classification, two main scenarios are the most widely adopted: (i) ***task-incremental*** (De Lange et al., 2021) and (ii) ***class-incremental*** (Belouadah et al., 2021). In both scenarios, the model has to learn new tasks, which are presented sequentially. Each incoming task provides the model with new, previously unseen classes, that need to be incorporated.

A common characteristic for both mentioned scenarios is the availability of task boundaries, which make the CL method aware that a new task is presented. The most relevant difference between the two scenarios is the availability of task labels, which provide the model with additional information on which task is being processed at the moment, during both training and inference. Specifically, a task-incremental scenario assumes the availability of task labels, whereas in class incremental learning, this information is not available.

It is worth stressing that the same data presented in different types of scenarios can yield significantly different results, since certain CL methods may be tailored for task-incremental scenarios, and as such the exploitation of task labels improve their

performance, while they may significantly suffer in class-incremental scenarios, where this information is not available.

The most widely adopted benchmark for class and task-incremental scenarios is split-MNIST (Kirkpatrick et al., 2016) consisting of 5 tasks. It leverages the original MNIST, separating it into five tasks, each containing two digits. In this class-incremental scenario, the model is not aware of what the current task is. It is only aware of the fact that it encountered a new task and needs to adjust itself. During the testing phase, the model is also not informed about which set of digits is currently provided, so the model has to classify one of the ten classes (digits 0-9). On the other hand, in the task-incremental variant of split-MNIST, the model is aware of which task is currently being presented, and only decides whether the image belongs to the first or the second class of the current task. This prediction, combined with information about the current task id, leads to the specific digit prediction.

Less commonly, certain scenarios relax the assumptions of class and task-incremental scenarios. Authors in Lomonaco et al. (2019) propose new scenarios that tackle the presentation of new training patterns of both known and unknown classes (New Instances and Classes - NIC), which include real-world challenges identified in some applications that were not considered before in continual learning scenarios. A recent trend in class-incremental learning is to adopt the repetition of previously encountered concepts in the learning scenario, which softens the disruption of previous knowledge (Cossu et al., 2022). Another example of constraint relaxation is domain-incremental scenario (Baker et al., 2023), where new distributions of the same classes are presented over time, as well as task-agnostic scenarios (Faber et al., 2022a, 2023), where neither task labels nor task boundaries are available, and reliance on external methods is necessary to detect task changes. Another interesting direction is that of online continual learning, which aims to learn directly from a data stream with shifting distribution (Carta et al., 2023; De Lange & Tuytelaars, 2021).

Overall, despite the widespread adoption of class-incremental and task-incremental settings, these studies highlight the opportunity for new continual learning scenarios that entail additional real-world complexities. Similarly, the same trend of extending CL towards challenging real-world conditions can be observed in studies that propose novel benchmarks/datasets.

The CLEAR benchmark Lin et al. (2021) proposes scenarios with a natural temporal evolution of visual concepts, entailing challenges similar to domain incremental learning settings. Another recent benchmark named CLiMB (Srinivasan et al., 2022) puts emphasis on multi-modal data, evaluating candidate CL models and learning algorithms on their forgetting, knowledge transfer, as well as their downstream low-shot transfer capability on both multimodal and unimodal tasks. The LECO benchmark (Lin et al., 2022) addresses the problem of refinement to ontologies with text data in continual learning, introducing a new ontology of "fine" labels that describe specific concepts and refine old ontologies of "coarse" labels that describe general concepts (e.g., dog breeds that refine the previously observed dog). The work in Ghunaim et al. (2023) proposes a practical real-time evaluation of continual learning, in which the stream does not wait for the model to complete training before revealing the next data. The authors perform experiments with the CLOC dataset (Cai et al., 2021), which contains 39 million time-stamped images with geolocation labels. Authors in Marsocci and Scardapane (2023) propose a novel dataset in the remote sensing domain, built as a combination of three previously introduced datasets containing images from airborne, satellite, and drone sources.

From a CL scenario viewpoint, in our study, we adopt class-incremental and task-incremental settings as they naturally fit the multi-task nature of our curriculum learning

scenario, where we are interested in assessing how effectively models incorporate diverse classes belonging to different datasets by increasing their difficulty over time. From a benchmark viewpoint, our work presents differences from the aforementioned studies. Specifically, differently than CLEAR and the studies in Ghunaim et al. (2023), Marsocci and Scardapane (2023), we attend to a multi-task overview where multiple heterogeneous datasets are analyzed, also resorting to a curriculum learning ordering of tasks. Moreover, unlike CLiMB, which puts emphasis on multi-modal data, and LECO, which analyzes textual data, our focus is strictly on the assessment of model longevity in the computer vision domain, which was not investigated before.

2.2 CL strategies

From a broad perspective, CL strategies belong to three main groups: using ***regularization***, ***dynamic architectures***, and ***rehearsal*** (also known as experience replay). In this paper, we consider popular and largely adopted CL strategies. We now describe each strategy and the rationale for its adoption in the benchmarks.

Regularization strategies influence the model weights adjustment process that takes place during model training in the attempt to preserve knowledge of previously learned tasks. The regularization strategies considered include Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2016), Learning without Forgetting (LwF), Synaptic Intelligence (SI) (Zenke et al., 2017), and Memory Aware Synapses (MAS) (Aljundi et al., 2018a). LwF (Li & Hoiem, 2017) aims at achieving output stability through knowledge distillation. When a new task is observed, the new model is incentivized to predict values that are close to the outputs of the model learned prior to this task. EWC (Kirkpatrick et al., 2016) and SI (Zenke et al., 2017) adopt a weighted quadratic regularization loss, which penalizes moving weights that are important for previous tasks. The EWC loss is based on the Fisher Information Matrix, which presents a higher computational complexity than the surrogate loss used in the SI method. Similarly, Memory Aware Synapses (MAS) (Aljundi et al., 2018a) estimates the cumulative importance of model weights as new tasks are encountered, penalizing changes to weights that are crucial for previously learned tasks. Shifting the focus on dynamic architectures, Progressive Neural Networks (PNN) is one of the first attempts to consider node expansion in a neural network architecture to accommodate different tasks (Rusu et al., 2016). CWRStar (Lomonaco et al., 2019) adapts weights exclusively for the last layer before the prediction layer, freezing all previous layers. AR1Star extends this capability by also tuning the representation layers (Lomonaco et al., 2019). Finally, rehearsal strategies considered include GDumb (Ameya Prabhu & Dokania, 2020), Replay (Rolnick et al., 2019), Gradient Episodic Memory (GEM) (David Lopez-Paz, 2017), and Average Gradient Episodic Memory (AGEM) (Chaudhry et al., 2019). GDumb Ameya Prabhu and Dokania (2020) is a greedy strategy that stores samples for all classes in a buffer, and uses them to iteratively retrain a model from scratch. It should be noted that GDumb was seen as a thought-provoking experiment showing that a performance close to state-of-the-art CL strategies could be achieved with a less sophisticated approach not specifically designed for CL problems. For this reason, it may be regarded as a baseline rather than a CL strategy. Replay Rolnick et al. (2019) follows a similar approach but stores a balanced number of samples per task, which are used to fine-tune previously trained models. A variant of this approach using generative models is pseudo-rehearsal (Shin

et al., 2017). GEM David Lopez-Paz (2017) is a fixed-size memory that stores a subset of old patterns and influences the loss function through inequality constraints. AGEM Chaudhry et al. (2019) is a revised version of GEM that performs averaging to increase efficiency.

The rationale for the adoption of the aforementioned strategies in our benchmark is that they are heterogeneous in terms of groups of approach, and are particularly prevalent in the CL community. They represent the foundations in the CL field, and are often used to assess the competitiveness of emerging CL methods with respect to consolidated and diversified approaches. Moreover, they are easy to use and favor reproducibility, thanks to publicly available tools such as the Avalanche library (Lomonaco et al., 2021).

2.3 CL evaluation protocol and metrics

The standard evaluation procedure applied in continual image classification assumes the availability of a set of tasks, each defined with a set of classes. The learning scenario consists of N tasks $T = t_1, t_2, \ldots, t_n$ where the model has to learn new tasks without forgetting previous tasks.

Metrics in continual learning usually focus on assessing the performance of a model (e.g., its accuracy) with respect to (at least one of) three crucial properties: *i)* performance on newly encountered tasks; *ii)* performance retention capabilities on previously learned tasks (i.e., the *ability to avoid or mitigate forgetting*); and *iii)* knowledge transfer from learned tasks to new ones (i.e., *the ability to generalize over newly occurring challenges*). The first works in CL proposed three metrics: average accuracy, backward transfer, and forward transfer to measure the above desiderata (David Lopez-Paz, 2017). However, in their original definition, only the performance of the model after learning all tasks was considered.

Instead, we consider model performances for all tasks and at all stages of the learning process, as understanding how performance changes before and after every task can provide several insights into the strengths and weaknesses of every model (Díaz-Rodríguez et al., 2018). For simplicity, we will be storing the partial model performance, measured as classification accuracy, in a matrix R whose entries $\mathsf{R}_{i,j}$ represent the accuracy on a given task j after learning task i.

Average Accuracy (ACC)—It measures the average accuracy of the model after learning each task, evaluating only the current and all previously learned tasks:

$$\mathsf{ACC} = \sum_{i \geq j}^{N} \mathsf{R}_{i,j} / (N(N-1)/2), \tag{1}$$

defined as the average performance over all tasks the model has seen so far.

Backward Transfer (BWT)—It measures the impact of learning new tasks on the performance of all previously learned tasks. Negative backward transfer indicates that learning a new task is harmful to the performance of previously learned tasks (this issue is known as forgetting):

$$\mathsf{BWT} = \sum_{i=2}^{N} \sum_{j=1}^{i-1} (\mathsf{R}_{i,j} - \mathsf{R}_{j,j}) / (N(N-1)/2), \tag{2}$$

defined as the average amount of forgetting presented by the model on the overall scenario.

Table 2 Overview of original datasets involved in our benchmarks

Dataset	Colors	Size	Classes	Available images	Accuracy
MNIST	BW	28×28	10	70,000	0.98
Omniglot	BW	105×105	1632	32,460	0.92
Fashion MNIST	BW	28×28	10	70,000	0.88
SVHN	RGB	32×32	10	630,420	0.85
CIFAR10	RGB	32×32	10	60,000	0.67
TinyImagenet	RGB	64×64	200	100,000	0.64

The datasets present heterogeneous characteristics, i.e., domains and technical quality. For TinyImagenet, we select the following classes: Egyptian cat; reel; volleyball; rocking chair; lemon; bullfrog; basketball; cliff; espresso; plunger. As for Omniglot, we select characters from the alphabet of the Magi. Accuracy refers to WideVGG9 performance on single datasets used to estimate task complexity

Forward Transfer (FWT)—It measures the impact of learned tasks on the performance of tasks learned in the future:

$$\mathsf{FWT} = \sum_{i<j}^{N} \mathsf{R}_{i,j}/(N(N-1)/2), \tag{3}$$

defined as the average model performance on yet unseen tasks.

3 Our benchmarks: M2I and I2M

In Sect. 1 we pointed out essential desiderata for continual learning benchmarks that are designed to reflect real-life environments and challenges. In the following, we further elaborate on each criterion, providing a rationale for its importance, and we describe how our benchmark tackles these challenges.

First, it is important to ***consider multiple heterogeneous tasks***. The rationale is that since continual learning models should adapt to new and unprecedented situations, as human beings usually act in real environments, they should be evaluated on sequences of heterogeneous tasks. While common benchmarks focus on homogeneous tasks, such as different classes of handwritten digits (e.g. as in splitMNIST), heterogeneous tasks have the advantage of reflecting more realistic cases where the model is challenged by unprecedented tasks with great diversity. To deal with multiple heterogeneous tasks, our benchmark leverages 6 largely-varying image classification datasets: MNIST (handwritten digits) [15], Omniglot (alphabets) (Lake et al., 2015), Fashion MNIST (clothing items) (Xiao et al., 2017), SVHN (street view house numbers) (Netzer et al., 2011), CIFAR10 (small real-world images) (Krizhevsky, 2009), and TinyImagenet (multi-domain large-scale real-world images) [34]. Each dataset is regarded as a task, resulting in a learning scenario with six tasks with heterogeneous characteristics. We provide more details about the datasets included and the preprocessing they underwent into the benchmark in Table 2.

Second, it is important to devise scenarios with ***varying quality and task complexity***, since an ideal model should present generalization capabilities dealing with easy, moderate, and difficult tasks at the same time, as found in the real world. Ideal scenarios should avoid simplistic sequences of tasks with high task similarity, and prefer introducing new tasks that are different enough from the previous one, thus challenging the

model in a significant way. This aspect should comprise having tasks on images varying in terms of visual and chromatic quality and difficulty of classification. Our benchmarks take this into consideration as they include very complex multi-domain real-world image classification such as TinyImagenet (harder classification), as well as handwritten digit recognition in MNIST (easier classification), and letter recognition in different alphabets in Omniglot (moderate difficulty). This choice of datasets creates ambitious but realistic challenges for CL strategies, allowing us to test their limitations.

Third, the hardness of each task is relative to the ordering in which the task is presented to the model. E.g. task ordering is important for us humans as we do not learn challenging new tasks from scratch but, instead, incrementally build up the necessary skills to perform these new tasks, leveraging a combination of skills learned in the past. We would require the same efficiency from a continual learner. Therefore, it is crucial to evaluate models ***learning on a direct or inverse curriculum***. The adoption of direct curriculum learning—learning on tasks of increasing complexity—in conventional machine learning research showcased that significant improvements in generalization can be achieved, increasing the speed of convergence of the training process (Bengio et al., 2009; Gao et al., 2022; Song et al., 2020).

When it comes to CL, however, direct and inverse curriculum learning are overlooked. Indeed, in the best cases, multiple random task orderings are provided in addition to a single task order. To properly consider curriculum learning, our benchmark considers curriculum learning by devising a task order according to their difficulty. To this end, we consider model performance as a proxy for task complexity. Specifically, we compute model accuracy on single datasets when considered in isolation. Table 2 shows the performance achieved by a WideVGG9 model with the different datasets considered in our study. These results create the conditions to define the ordering of datasets as tasks in our M2I and I2M benchmarks. The scenario starts with MNIST (black & white handwritten digits), which is regarded as an easy task. The following tasks are Omniglot (alphabets) and Fashion MNIST (clothing items), which present a spike of complexity compared to MNIST. Subsequently, SVHN (street view house numbers) brings real-world complexity by introducing images gathered from cameras with colors. CIFAR10 presents the same challenges of real-world colored images and extends them with more challenging patterns encountered in complex objects. Finally, the highest level of complexity is provided by multi-domain large-scale images from TinyImagenet. In addition to the direct curriculum direction where tasks are ordered as described (from MNIST to TinyImageNet, aka M2I), we also cover the opposite case of decreasing order of difficulty (from TinyImageNet to MNIST, aka I2M).

Fourth, ***rigorous way to measure generalization and forgetting***. The most important aspect of continual machine learning is to design strategies and models that are able to incorporate new tasks during their lifespan, without forgetting previous tasks. Metrics such as BWT and FWT are introduced for this reason, see Sect. 2.3 However, they can be cumbersome to interpret or lose their meaning, depending on the learning setting at hand. For instance, FWT is ill-defined in a class-incremental scenario since the model will never predict classes that were never presented before. Another example is that of multi-dataset benchmarks where tasks contain a varying number of classes. Specifically, tasks with a reduced number of classes will exhibit a random performance that is higher (e.g., 0.5 for 2 classes) than tasks with a higher number of classes (e.g., 0.1 for 10 classes). As results are generally aggregated (i.e., averaged) across tasks (Aljundi et al., 2018b; Ebrahimi et al., 2020; Li & Hoiem, 2017; Lange et al., 2022; Mallya & Lazebnik, 2017), CL metrics will be hard to interpret due to a different reference point for random performance. Ideal

benchmarks should take these aspects into consideration to make sure that the calculation and the interpretation of results are correct.

To consider this aspect, we designed each task in our benchmark to contain 10 classes. In the case of MNIST and Fashion MNIST, SVHN, and CIFAR10, we use all classes. In the case of TinyImageNet and Omniglot, we select 10 classes. For TinyImagenet, we use Egyptian cat; reel; volleyball; rocking chair; lemon; bullfrog; basketball; cliff; espresso; plunger. As for Omniglot, we select classes corresponding to characters from the Alphabet of the Magi. This setting allows us to preserve a high interpretability of all the resulting metric values overcoming the limitation of tasks with an imbalanced number of classes, where interpretability can be lost. Furthermore, to deal with class imbalance, we align the size of majority classes to that of minority classes. By doing so, we isolate the learning setting and avoid typical issues that arise in imbalanced learning, which might undermine the analysis of the final results.

Fifth, ***exactly reproducible out-of-the-box***. Many benchmarks are not reproducible due to the lack of precise details on model configurations and experimental settings. This issue is exacerbated when the code is unavailable and it is required to implement the scenario and the evaluation scheme from scratch. In other cases, when the code is available, it is not general enough to be leveraged in different settings, e.g. when comparing with the latest models and strategies. To this end, our benchmark is implemented on top of Avalanche (Lomonaco et al., 2021)—the state-of-the-art open-source library for CL. This choice ensures the reproducibility of the experiments and paves the way for the adoption and extension of the benchmark for future research. The code for our benchmarks is publicly available at the following repository URL: https://github.com/lifelonglab/M2I_I2M_benchmark.

It is worth noting that, despite the adoption of class and task-incremental scenarios, our benchmark can be easily extended to different emerging scenarios, such as incremental data learning (De Lange & Tuytelaars, 2021), in task-free and task-agnostic settings. To this end, a procedure can select a subset of datasets according to their complexity. Each batch may present multiple tasks, i.e. data from all datasets in the subset. After a number of batches, the procedure can remove the less (more) complex dataset and add a new more (less) complex dataset according to the curriculum ordering defined in our benchmark. It is conceivable that this procedure would let some tasks disappear while letting others appear, as frequently observed in online learning, but in increasing (or decreasing) order of difficulty, giving place to a curriculum learning scenario.

4 Experiments and discussion

We carry out experiments involving both the *task-incremental* and *class-incremental* CL scenario types described in Sect. 2.1, the CL strategies devised in Sect. 2.2, and the CL evaluation protocol and metrics defined in Sect. 2.3. We run an exhaustive series of experiments on our proposed M2I and I2M benchmarks for CL, resulting in 156 complete experiments (considering M2I and I2M with 14 CL strategies, 2 scenario types - Class-incremental and Task-incremental, and 3 model backbones) and 936 runs (model training and evaluation). We aim to answer the following research questions:

- *RQ1)* Do our benchmarks provide challenging scenarios for state-of-the-art CL strategies as discussed in Sect. 2.2? That is, are these strategies still as accurate and robust

w.r.t the metrics defined in Sect. 2.3 as originally introduced in their papers when exposed to M2I and I2M?

- *RQ2)* Can state-of-the-art CL strategies leverage direct and indirect curriculum task ordering to maximize their backward and forward transfer? Alternatively, do different task orderings with varying task complexity have an impact on the final performance?

We first detail the experimental setup of our experiments and then provide an in-depth discussion of the results gathered. For the curious reader, the short answer to both questions is that, overall, the state-of-the-art models underperform when executed on our challenging benchmarks, despite many of these models were supposed to be robust to catastrophic forgetting and multiple tasks.

4.1 Experimental setup

As mentioned in Sect. 3, our benchmark provides multiple heterogeneous tasks with varying quality and task complexity. For instance, 3 of the 6 tasks contain black and white images, whereas the remainder contain colored images. Moreover, the image size varies across all tasks. There may be different ways to deal with different image channel types and sizes, which can have an impact on the final performance. However, we recognize that finding the optimal solution is an open challenge for researchers working with our benchmark, and it is out of the scope of this paper. For simplicity, for image sizes, we adopt the most frequently adopted approach, which consists of resizing all images to the same size (64×64). To deal with different image channels, we consider the largest number of channels (RGB) for all tasks (3) and replicate the single-channel encountered in BW images to all 3 channels. We recall that, in order to provide a rigorous way to measure generalization and forgetting, we balance class sizes by taking 500 images from each of them for both the training and evaluation phases. By doing so, we isolate possible issues deriving from class imbalance from our evaluation.

Network architecture. We leverage three commonly used model backbones in CL with different parameter sizes as to measure the effect of overparametrization w.r.t. our performance metrics in CL. Each network architecture is being used across all strategies. We employ a Wide VGG9 (Simonyan & Zisserman, 2014) as a smaller neural network (4.5M parameters), EfficientNet-b1 (Tan & Le, 2019) as a mid-size alternative (7.8M parameters), and ResNet34 (He et al., 2016) as a large-size state-of-the-art model backbone (63.5M parameters). The hyperparameter configuration used in the experiments is: *{ epochs=50, learning_rate=0.001, momentum=0.9 }*. Optimization takes place through Stochastic Gradient Descent (SGD) using the Cross-Entropy loss. We experimented with different negative powers of 10 for the configuration of the learning rate as suggested in Bengio (2012), For the number of epochs, we experimented with similar values to those reported in the original publications of CL strategies (Aljundi et al., 2018b; Rolnick et al., 2019). Preliminary experiments showed that different configurations did not provide a significant difference in terms of performance metric values. For PNN, our setting differs from other strategies due to the fact that the Avalanche implementation provides support only for fully connected layers, leading to the impossibility of pairing the learning strategy with specific CNN model backbones (WideVGG9, EfficientNet, ResNet34). To this end, we matched the number of fully connected layers with those of our adopted CNN model architectures, i.e., 9,

Table 3 Experimental results (Wide-VGG99—M2I) in terms of average performance (and rank) for all CL strategies grouped by type (from top to bottom: regularization, rehearsal, architectural, and baseline) in two learning settings (class-incremental, task-incremental)

	Class-incremental		Task-incremental		
	ACC	BWT	ACC	BWT	FWT
EWC	0.220 (8)	– 0.271	0.395 (10)	– 0.224	0.102
LwF	0.222 (7)	– 0.270	0.349 (12)	– 0.083	0.096
MAS	0.215 (9)	– 0.260	0.440 (7)	– 0.212	0.100
SI	0.206 (11)	– 0.255	0.419 (8)	– 0.219	0.107
AGEM	0.188 (12)	– 0.241	0.472 (6)	– 0.161	0.101
GEM	0.572 (3)	– 0.074	0.613 (5)	– 0.053	0.099
GenerativeReplay	0.558 (4)	– 0.142	0.616 (4)	– 0.133	0.102
Replay	0.755 (2)	– 0.038	0.730 (3)	– 0.086	0.101
CWRStar	0.324 (5)	– 0.044	0.356 (11)	– 0.019	0.094
PNN			0.091 (14)	0.000	0.093
Naive	0.213 (10)	– 0.261	0.411 (9)	– 0.228	0.093
GDumb	0.304 (6)	– 0.040	0.235 (13)	– 0.071	0.092
Cumulative	0.868 (1)	0.004	0.819 (2)	0.012	0.102
MSTE			0.872 (1)	0.000	0.100

24, and 34 layers, in an attempt to simulate their capacity. We also note that our PNN results are computed exclusively in the task-incremental setting since task identifiers are not available in the class-incremental setting.

CL strategies. In addition to the state-of-the-art CL strategies covered by our experiments and described in Sect. 2.2, we adopt three additional baseline approaches that loosely correspond to lower and upper bound model performance:

- *Naive (fine-tuning)*: The model is incrementally fine-tuned without considering any mechanism to preserve past knowledge, which, in principle, should yield a high degree of forgetting. This strategy allows us to compare the performance (in terms of accuracy) and forgetting (in terms of backward transfer) of smarter CL strategies.
- *Cumulative*: New data is accumulated as it comes, and the model is retrained using all available data. The rationale for this baseline is to simulate upper-bound performance assuming full knowledge of the data, and unlimited computational resources to deal with stored data (storage) and model retraining (time). Cumulative can also be regarded as a variant of Replay with unlimited memory. This baseline is interesting since it allows us to estimate the accuracy that could be achieved at a much higher computational cost.
- *Multiple Single-Task Expert (MSTE)*: A new model is created as soon as a new task is presented in the scenario. It can be regarded as a way to simulate upper-bound model performance despite a few unrealistic assumptions, such as unlimited computational resources to deal with additional models and availability of task identifiers (which make this baseline suitable only for Task-incremental scenarios).

Table 4 Experimental results (Wide-VGG99—I2M) in terms of average performance (and rank) for all CL strategies grouped by type (from top to bottom: regularization, rehearsal, architectural, and baseline) in two learning settings (class-incremental, task-incremental)

	Class-incremental		Task-incremental		
	ACC	BWT	ACC	BWT	FWT
EWC	0.190 (9)	− 0.202	0.340 (7)	− 0.161	0.126
LwF	0.216 (6)	− 0.238	0.262 (11)	− 0.105	0.088
MAS	0.223 (5)	− 0.241	0.342 (6)	− 0.148	0.130
SI	0.205 (7)	− 0.223	0.323 (9)	− 0.170	0.134
AGEM	0.175 (10)	− 0.180	0.345 (5)	− 0.094	0.131
GEM	0.297 (4)	− 0.031	0.269 (10)	0.005	0.117
GenerativeReplay	0.351 (3)	− 0.204	0.423 (4)	− 0.139	0.136
Replay	0.550 (2)	− 0.047	0.571 (3)	− 0.061	0.135
CWRStar	0.079 (12)	− 0.037	0.202 (12)	− 0.011	0.107
PNN			0.101 (14)	0.000	0.073
Naive	0.199 (8)	− 0.215	0.334 (8)	− 0.160	0.131
GDumb	0.155 (11)	− 0.052	0.147 (13)	0.000	0.085
Cumulative	0.735 (1)	0.012	0.663 (2)	0.018	0.120
MSTE			0.695 (1)	0.000	0.100

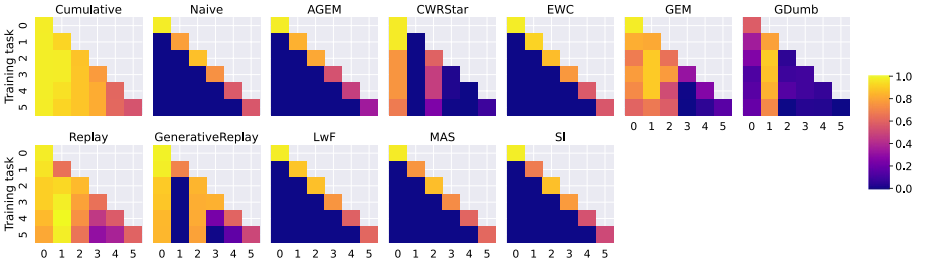

Fig. 2 Experimental results (Wide-VGG9—M2I—Class-incremental) in terms of disaggregated performance (ACC) on single tasks after learning previous tasks

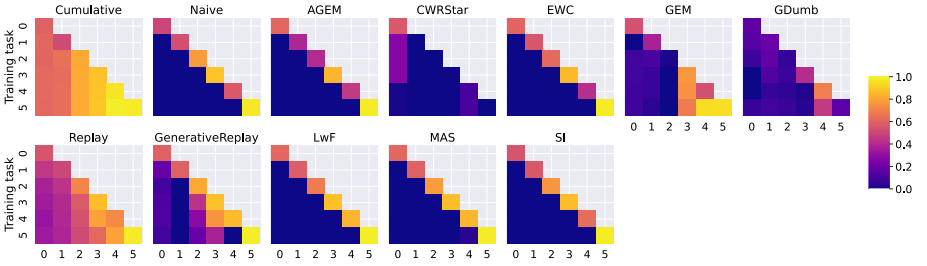

Fig. 3 Experimental results (Wide-VGG9—I2M—Class-incremental) in terms of disaggregated performance (ACC) on single tasks after learning previous tasks

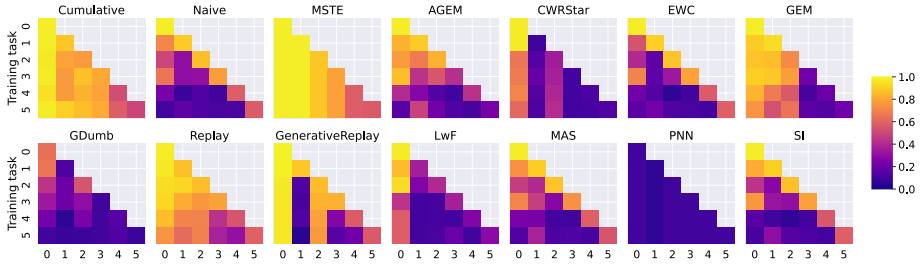

Fig. 4 Experimental results (Wide-VGG9—M2I—Task-incremental) in terms of disaggregated performance (ACC) on single tasks after learning previous tasks

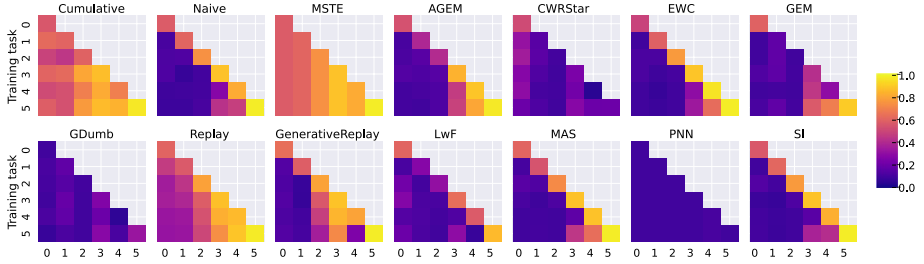

Fig. 5 Experimental results (Wide-VGG9—I2M—Task-incremental) in terms of disaggregated performance (ACC) on single tasks after learning previous tasks

Table 5 Experimental results (EfficientNet—M2I) in terms of average performance (and rank) for all CL strategies grouped by type (from top to bottom: regularization, rehearsal, architectural, and baseline) in two learning settings (class-incremental, task-incremental)

	Class-incremental		Task-incremental		
	ACC	BWT	ACC	BWT	FWT
EWC	0.180 (9)	− 0.222	0.272 (11)	− 0.199	0.106
LwF	0.165 (11)	− 0.205	0.240 (12)	− 0.124	0.096
MAS	0.187 (7)	− 0.190	0.276 (10)	− 0.198	0.104
SI	0.184 (8)	− 0.231	0.277 (9)	− 0.205	0.099
AGEM	0.179 (10)	− 0.224	0.328 (7)	− 0.171	0.091
GEM	0.312 (5)	− 0.056	0.420 (5)	− 0.043	0.096
GenerativeReplay	0.517 (3)	− 0.143	0.551 (4)	− 0.102	0.097
Replay	0.655 (2)	− 0.041	0.605 (3)	− 0.098	0.102
CWRStar	0.326 (4)	− 0.014	0.367 (6)	0.001	0.092
PNN			0.091 (14)	0.000	0.100
Naive	0.205 (6)	− 0.257	0.290 (8)	− 0.220	0.103
GDumb	0.029 (12)	− 0.007	0.099 (13)	0.000	0.100
Cumulative	0.834 (1)	0.003	0.656 (2)	0.025	0.101
MSTE			0.784 (1)	0.000	0.100

Table 6 Experimental results (EfficientNet—I2M) in terms of average performance (and rank) for all CL strategies grouped by type (from top to bottom: regularization, rehearsal, architectural, and baseline) in two learning settings (class-incremental, task-incremental)

	Class-incremental		Task-incremental		
	ACC	BWT	ACC	BWT	FWT
EWC	0.197 (7)	− 0.182	0.299 (7)	− 0.169	0.100
LwF	0.203 (5)	− 0.182	0.241 (11)	− 0.072	0.091
MAS	0.178 (9)	− 0.164	0.284 (9)	− 0.148	0.105
SI	0.193 (8)	− 0.177	0.302 (6)	− 0.143	0.105
AGEM	0.157 (10)	− 0.134	0.289 (8)	− 0.101	0.122
GEM	0.218 (4)	− 0.052	0.276 (10)	− 0.052	0.119
GenerativeReplay	0.313 (3)	− 0.179	0.390 (4)	− 0.094	0.124
Replay	0.417 (2)	− 0.019	0.438 (3)	− 0.058	0.099
CWRStar	0.053 (11)	− 0.035	0.171 (12)	− 0.020	0.103
PNN			0.096 (14)	0.000	0.095
Naive	0.202 (6)	− 0.193	0.305 (5)	− 0.154	0.108
GDumb	0.029 (12)	0.000	0.100 (13)	0.000	0.092
Cumulative	0.606 (1)	0.021	0.596 (1)	0.015	0.118
MSTE			0.556 (2)	0.000	0.100

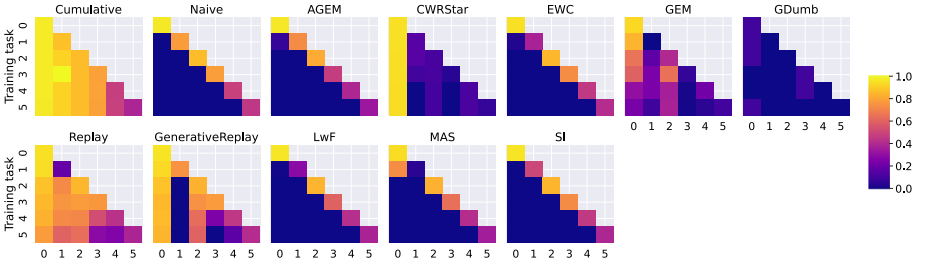

Fig. 6 Experimental results (EfficientNet—M2I—Class-incremental) in terms of disaggregated performance (ACC) on single tasks after learning previous tasks

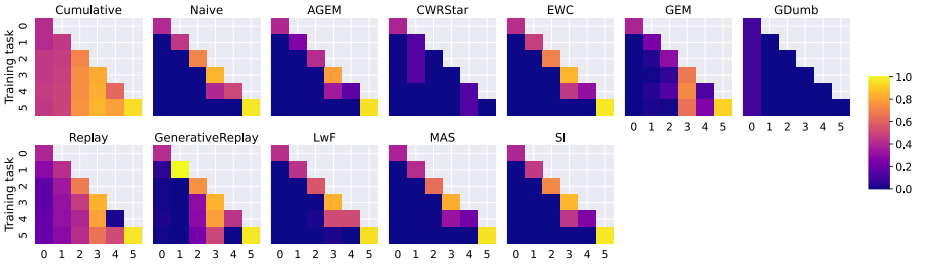

Fig. 7 Experimental results (EfficientNet—I2M—Class-incremental) in terms of disaggregated performance (ACC) on single tasks after learning previous tasks

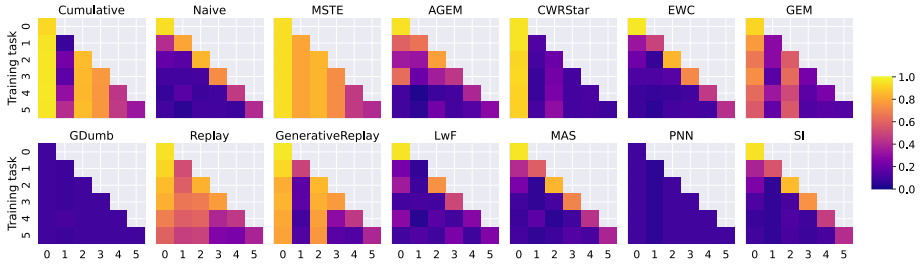

Fig. 8 Experimental results (EfficientNet—M2I—Task-incremental) in terms of disaggregated performance (ACC) on single tasks after learning previous tasks

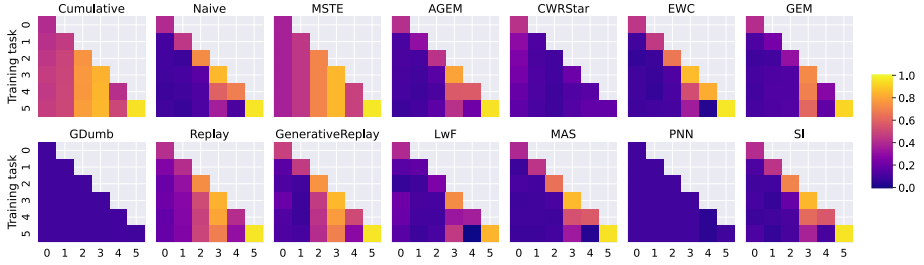

Fig. 9 Experimental results (EfficientNet—I2M—Task-incremental) in terms of disaggregated performance (ACC) on single tasks after learning previous tasks

Table 7 Experimental results (ResNet—M2I) in terms of average performance (and rank) for all CL strategies grouped by type (from top to bottom: regularization, rehearsal, architectural, and baseline) in two learning settings (class-incremental, task-incremental)

	Class-incremental		Task-incremental		
	ACC	BWT	ACC	BWT	FWT
EWC	0.175 (10)	− 0.222	0.314 (9)	− 0.213	0.103
LwF	0.174 (12)	− 0.218	0.257 (13)	− 0.124	0.100
MAS	0.195 (8)	− 0.232	0.300 (11)	− 0.200	0.099
SI	0.182 (9)	− 0.229	0.301 (10)	− 0.206	0.098
AGEM	0.208 (7)	− 0.228	0.464 (6)	− 0.179	0.093
GEM	0.412 (4)	− 0.155	0.601 (5)	− 0.088	0.099
GenerativeReplay	0.447 (3)	− 0.162	0.632 (3)	− 0.102	0.105
Replay	0.630 (2)	− 0.087	0.612 (4)	− 0.090	0.093
CWRStar	0.331 (6)	− 0.006	0.403 (7)	0.013	0.097
PNN			0.089 (14)	0.000	0.097
Naive	0.175 (11)	− 0.223	0.322 (8)	− 0.198	0.089
GDumb	0.357 (5)	− 0.033	0.294 (12)	− 0.044	0.099
Cumulative	0.788 (1)	0.003	0.750 (2)	0.007	0.099
MSTE			0.784 (1)	0.000	0.100

Table 8 Experimental results (ResNet—I2M) in terms of average performance (and rank) for all CL strategies grouped by type (from top to bottom: regularization, rehearsal, architectural, and baseline) in two learning settings (class-incremental, task-incremental)

	Class-incremental		Task-incremental		
	ACC	BWT	ACC	BWT	FWT
EWC	0.191 (8)	− 0.204	0.300 (8)	− 0.117	0.104
LwF	0.188 (10)	− 0.200	0.221 (12)	− 0.076	0.094
MAS	0.196 (7)	− 0.180	0.308 (7)	− 0.138	0.118
SI	0.189 (9)	− 0.201	0.288 (10)	− 0.142	0.108
AGEM	0.221 (5)	− 0.157	0.375 (6)	− 0.110	0.145
GEM	0.264 (4)	− 0.086	0.388 (5)	− 0.073	0.131
GenerativeReplay	0.313 (3)	− 0.131	0.424 (4)	− 0.107	0.135
Replay	0.419 (2)	− 0.074	0.443 (3)	− 0.058	0.080
CWRStar	0.102 (12)	− 0.026	0.228 (11)	− 0.004	0.097
PNN			0.095 (14)	0.000	0.085
Naive	0.182 (11)	− 0.191	0.291 (9)	− 0.114	0.104
GDumb	0.204 (6)	− 0.031	0.180 (13)	− 0.017	0.101
Cumulative	0.614 (1)	0.006	0.562 (1)	0.004	0.104
MSTE			0.552 (2)	0.000	0.100

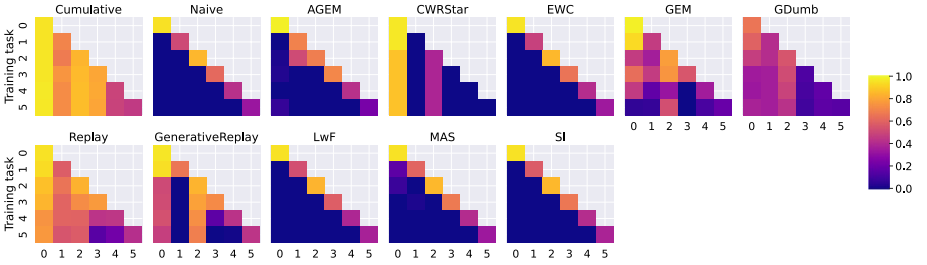

Fig. 10 Experimental results (ResNet—M2I—Class-incremental) in terms of disaggregated performance (ACC) on single tasks after learning previous tasks

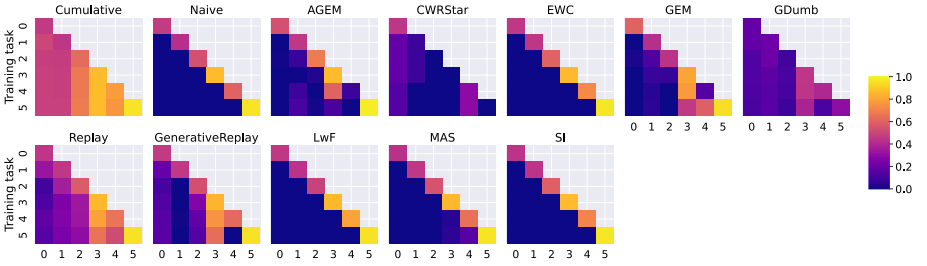

Fig. 11 Experimental results (ResNet—I2M—Class-incremental) in terms of disaggregated performance (ACC) on single tasks after learning previous tasks

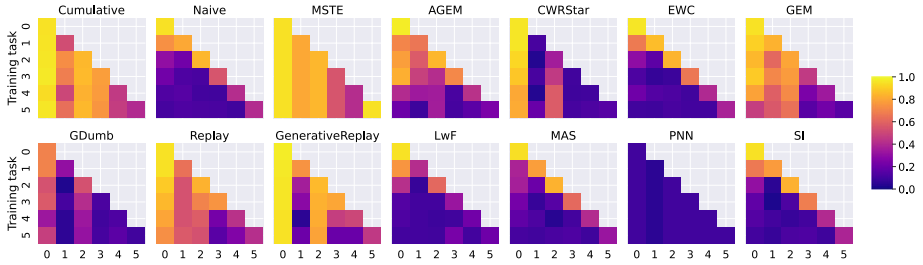

Fig. 12 Experimental results (ResNet—M2I—Task-incremental) in terms of disaggregated performance (ACC) on single tasks after learning previous tasks

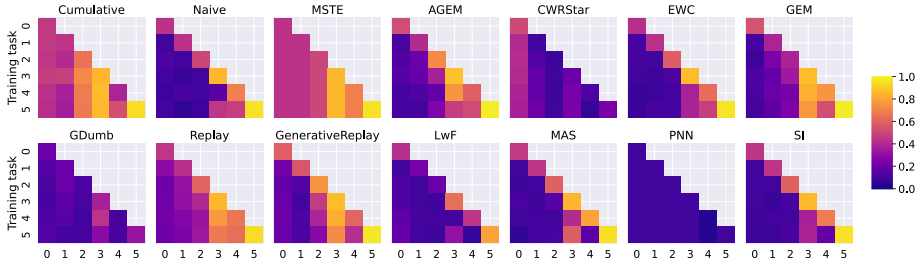

Fig. 13 Experimental results (ResNet—I2M—Task-incremental) in terms of disaggregated performance (ACC) on single tasks after learning previous tasks

4.2 Discussion: RQ1

We present our results both as aggregated metrics computed after all the tasks have been learned in Tables 3, 4, 5, 6, 7 and 8 as well as disaggregated accuracy results evaluating every task past task after learning a new one (as entries in the matrix R, see Sect. 2.3) as heatmaps shown in Figs. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 and 13. These heatmaps allow us to understand at a finer grain what are the failure modes of a strategy and whether certain tasks are harder than another. We now discuss four different settings, comprising either a class-incremental or task-incremental scenario and two task orderings (M2I or I2M). We start from employing the VGG9 model backbone.

4.2.1 VGG9

For both M2I and I2M (see Tables 3, 4), aggregated ACC, BWT and FWT (only for task-incremental scenarios, see Sect. 3) are disappointing for all strategies discussed in Sect. 2.2. They do not achieve a positive backward transfer, and all highlight a systematic catastrophic forgetting, while the forward transfer floats around the chance level (10%[2]). Staple strategies such as LwF, MAS, and SI are generally comparable with the Naive strategy (i.e., just applying fine-tuning).

[2] We remark that this trend is easy to spot and understand in our benchmarks as all tasks sports only 10 classes.

We inspect rankings over the ACC score to see if any notable trend manifests between different scenario types. We found that AGEM is very weak in the class-incremental setting (ranked 12 and 10, respectively), but quite robust in the task-incremental setting (ranked 6 and 5). Surprisingly, GEM seems robust in both settings (ranked 3, 4, 5, and 10 across the four settings). CWRStar achieves a moderately high position in the class-incremental setting (ranked 5 in the ranking for M2I). While its ranking is surprisingly high, we remark that the raw performance is clearly unsatisfactory when considered in absolute terms in this context. A much lower ranking is observed in the task-incremental setting despite the slight improvement in its performance. Overall, CWRStar appears ineffective in preventing catastrophic forgetting across all tasks in our settings, and it appears that it just focuses on memorizing the first task. We also observe that PNN presents a remarkably low performance, which consistently ranks 14 throughout both scenarios.

The method that seems to be the least prone to forgetting is Replay. This result is surprising since the total memory size chosen for the replay buffer in the experiment is just 200 samples. It is also interesting to observe that Replay presents a performance that is quite close to Cumulative (an 8–18% decrease in accuracy across the four mentioned settings) using a fraction of available data (less than 1%). Surprisingly, GenerativeReplay performs quite poorly when compared to standard Replay, although it outperforms many other strategies and presents a rank of 3 or 4 in all scenarios. As expected, Cumulative presents the best performance across three out of four learning settings. MSTE outperforms Cumulative in the I2M scenario - Task-incremental setting. However, both strategies should be seen as an unrealistic upper bound, since Cumulative assumes that infinite memory and training time are allowed for the model, whereas MSTE assumes the ability to add new models continuously. On a different note, the positive results confirm that the scenarios designed in our benchmark are reasonable and can be, in principle, learned by the model but current CL strategies. While these have shown to be reliable in conventional CL scenarios, they are not bulletproof and present limited robustness when exposed to more complex scenarios, such as our M2I (**RQ1**).

Observing the heatmaps in Figs. 2, 3, 4 and 5 allows us to zoom in and pinpoint the performance drops of different strategies on specific tasks. In this context, observing a decreasing performance on previously learned tasks is a clear manifestation of forgetting. Results are quite negative for M2I in the class-incremental setting (see Fig. 2). GEM preserves a good performance until the third task is presented and then dramatically drops in the following tasks due to their increasing complexity. GDumb presents a high performance on the second task throughout the entire scenario but an unsatisfactory performance on all other tasks. This behavior likely depends on the fact that it is possible to learn the second task (Omniglot) with a limited number of samples, whereas this is too difficult for all other tasks. All other strategies, except Replay and Cumulative, struggle to preserve the knowledge of previous tasks and are successful at learning the last task exclusively, as evidenced by the very low-performance scores. GenerativeReplay presents a slightly different behavior, where the performance on certain tasks (0 and 2) is preserved, while the method appears unreliable on other tasks (1, 3, 4, and 5). The results for M2I in the task-incremental setting (see Fig. 4) showcase a higher overall performance, with lower forgetting than the class-incremental setting. For instance, it can be observed that MAS and SI is able to preserve much more knowledge for some tasks, while its forgetting was rather drastic in the class-incremental setting.

For I2M in the class-incremental setting (see Fig. 3), a similar behavior to the class-incremental counterpart of M2I can be observed, with drastic forgetting, which is even

worse than the M2I scenario. As for I2M in the task-incremental setting (see Fig. 5), it is also interesting to observe worse results than in the M2I task-incremental setting. This is also counterintuitive, as successfully learning a harder task should provide the model with enough knowledge not to perform so poorly on much simpler tasks that, as MNIST, might require learning only simple edge detectors. We conjecture that this behavior might depend on the fact that once a model is presented with very different but complex tasks earlier in the scenario (e.g., ImageNet and CIFAR10) it might have a harder time learning to abstract useful features for simpler tasks later.

4.2.2 EfficientNet

Results on M2I and I2M (see Tables 5, 6) show that the Naive strategy achieves a performance that is close to some of the CL strategies (e.g. MAS, SI, EWC) but is significantly inferior to top performing strategies (Replay, Cumulative, MSTE). When comparing class-incremental and task-incremental settings for M2I, some methods appear significantly more robust in the latter, with a simultaneous increase in their performance and position in the ranking.[3] This is the case for AGEM (ranked 10 and 7, respectively). GEM preserves its ranking (5) in both settings, while improving its performance in the task-incremental setting. For I2M, comparing class and task-incremental settings, the same phenomenon can be observed for two methods: AGEM (ranked 10 and 8, respectively), SI (ranked 8 and 6, respectively). Two methods preserve their ranking in both settings: MAS and EWC (ranked 9 and 7, respectively). Some strategies present a rather stable behavior in the two learning settings, since they appear to preserve their ranking. For M2I, this is the case for GDumb, LwF, GenerativeReplay, and Replay. For I2M, this phenomenon applies to CWRStar, GDumb, GenerativeReplay, and Replay. Similarly to results obtained with VGG9, Cumulative showcases the best performance in three out of four learning settings, while MSTE is the top-ranked strategy in one out of four cases. Therefore, in absolute terms, the performance of informed strategies can be regarded as unsatisfactory, as it appears significantly lower than Cumulative and MSTE. This result suggests that even increasing the parametrization of the backbone model does not provide these staple CL strategies to significantly improve over our simpler baselines in complex benchmarks such as our M2I and I2M (**RQ1**).

Shifting our focus to the heatmaps in Figs. 6, 7, 8 and 9, we are able to analyze in detail the forgetting of the different strategies throughout the experimental scenario.

Figure 6 shows a vast amount of forgetting across all strategies. Some exceptions can be sparsely observed. For instance, MAS preserves its performance on task 0 after learning task 1, before dropping to values that are close to zero for previously encountered tasks. CWRStar preserves a remarkably high performance on the first task, but a very limited ability to incorporate new tasks. This result is in contrast with what was observed with VGG9, where the performance on the first task was preserved but decaying as new tasks are presented. This phenomenon may depend on the number of layers involved in the model backbone since EfficientNet is a much larger model, and the weight adaptation strategy used in CWRStar exclusively involves the last layer. Interestingly, we observe that GenerativeReplay is unable to preserve task 1, while it proves to be better than Replay in

[3] It is important to track both aspects, since the task-incremental setting is fundamentally easier than class-incremental, and observing only the absolute performance of the methods is not indicative of an improvement.

remembering task 0. However, Replay presents a more diffused ability to preserve all tasks, as shown by its higher average performance. Moving to I2M in the class-incremental setting (see Fig. 7), a noteworthy result is GEM preserving knowledge of task 3 throughout the entire scenario, while not being able to preserve its performance on the other tasks. In this setting, CWRStar is fundamentally unable to learn any of the tasks presented in the scenario.

Interestingly, task similarity between two tasks, manifested by positive backward transfer, allows for improvement on previously learned tasks in some instances. In Fig. 8, for instance, learning task 4 is, in some cases, beneficial for the model's performance on task 1, as observed for MAS, GEM, and Naive.

In the task-incremental setting (see Fig. 9), GEM presents a similar behavior to that observed in class-incremental on task 2, but also preserves knowledge of task 0 throughout the entire scenario, whereas the performance on other tasks is fundamentally sub-optimal. In this setting, however, CWRStar behaves as in the M2I class-incremental setting, i.e., the performance on task 0 is preserved throughout the entire scenario.

4.2.3 ResNet

Results on M2I and I2M (see Tables 7, 8) show that the Naive strategy is relatively close to some of the CL strategies (e.g., EWC, SI, LWF) but is clearly less effective than other strategies (Replay, GenerativeReplay, Cumulative, MSTE). As observed with other model backbones, a comparison of class-incremental and task-incremental settings for M2I, highlights that some methods are more accurate in the latter and achieve a higher ranking. This phenomenon is observed for AGEM (ranked 7 and 6, respectively), Naive (11 and 8, respectively), and EWC (ranked 10 and 9, respectively). Similarly, in the I2M scenario, some of the strategies present an improved performance in the task-incremental setting: Naive (ranked 11 and 9, respectively), and CWRStar (ranked 12 and 11, respectively). Two strategies preserve their ranking across the two settings: EWC and MAS (ranked 8 and 7, respectively). Two exceptions can be identified: LwF and GDumb, which exhibit a lower ranking in the task-incremental setting (12, and 13, respectively) when compared to class-incremental (10, and 6, respectively). Strategies with stable behavior include Replay, GenerativeReplay, MSTE, and Cumulative in the two learning settings, which appears to preserve its ranking.

Cumulative confirms top-performing scores in three out of four settings. MSTE achieves the highest rank in the M2I - task-incremental setting. Overall, we observe that, unexpectedly, adopting a larger model such as ResNet did not achieve significantly better results than other model architectures with a reduced number of parameters and did not alter the considerations drawn in our discussion with other models.

Similarly to what we observed with other model backbones, Figs. 10, 11, 12 and 13 show that all strategies are largely affected by forgetting. However, it is worth noting that some strategies preserve a certain degree of knowledge for specific tasks. Notable examples include CWRStar (see Fig. 10), which preserves full knowledge of task 0 across the entire learning scenario. In the same scenario, GenerativeReplay is quite effective in knowledge preservation for tasks 0 and 2. Replay presents a wider retention capability that extends to multiple tasks when compared to CWRStar and GenerativeReplay. GDumb effectively incorporates and preserves knowledge of tasks 0, 1, and 2, whereas it struggles to learn the following tasks. All other strategies are essentially ineffective at retaining previously learned tasks, as shown by the large amount of forgetting.

Observing Fig. 11, the most interesting result is that of GenerativeReplay, which appears ineffective in retaining knowledge of more difficult tasks, even when they are presented as the initial ones in the I2M scenario (tasks 0, 1, and 2). It is noteworthy that for Replay, we observe an improvement on task 0 after training task 3, which suggests the ability of the model and strategy to leverage tasks similarity.

An interesting takeaway in the task-incremental M2I setting (Fig. 8) is that AGEM is less affected by forgetting, which is particularly visible on tasks 0, 1, and 2. A less visible but similar pattern can be observed with LwF, MAS, and SI.

Moving to the task-incremental I2M scenario (Fig. 9), we observe that results are generally disappointing, with diffused forgetting. However, Replay and GenerativeReplay present an interesting behavior, where tasks encountered in the second half of the scenario are better retained. This may suggest that Replay and GenerativeReplay are able to deal with easier tasks better, even though they are introduced later in the scenario.

In general, we observe that ResNet presents a high degree of forgetting, as noticed with other model backbones. This phenomenon is emphasized in our heatmaps in Figs. 10, 11, 12 and 13. Therefore, in absolute terms, the performance of informed strategies can be regarded as unsatisfactory, as it appears significantly lower than Cumulative. This result suggests that even increasing the parametrization of the backbone model does not provide these staple CL strategies to significantly improve over our simpler baselines in complex benchmarks such as our M2I and I2M (**RQ1**).

4.2.4 Summary: RQ1

In summary, results observed across the two learning settings (class-incremental, task-incremental) in the two presentation orders (M2I, I2M) show unsatisfactory performance for all learning strategies and that catastrophic forgetting is a real burden for many of the covered methods.

Considering that the results observed are inferior when compared to what is commonly reported in continual learning research, we can argue that our benchmark provides more challenging conditions for the CL strategies. It is noteworthy that forgetting in CL strategies is also observed in perceptually similar tasks (e.g., MNIST, Omniglot, SVHN), as evident in our heatmaps. This behavior is indicative of the objective lack of robustness presented by CL strategies as they are exposed to tasks from different datasets. The five desiderata described in Sect. 3 and adopted to design our benchmarks set up a higher standard for the evaluation of CL strategies, and will hopefully stimulate the design and implementation of new, more robust strategies.

4.3 Discussion: RQ2

In this subsection, we focus on the assessment of the ability of CL strategies to leverage curriculum task ordering to maximize their performance when exposed to our benchmarks devised in Sect. 3.

To answer this question, we start by analyzing results in Tables 3, 4, 5, 6, 7 and 8, which show metric values for the two scenarios: M2I (direct curriculum learning) and I2M (inverse curriculum learning). Almost all methods present a better performance in the curriculum learning setting (M2I) when compared with the inverse curriculum setting (I2M). Comparing values in Tables 3 and 4, significant examples for VGG9 include Replay (from 0.550 to 0.755 in class-incremental and from 0.571 to 0.730 in task-incremental),

GenerativeReplay (from 0.351 to 0.558 in class-incremental and from 0.423 to 0.616 in task-incremental), and GEM (from 0.297 to 0.572 in class-incremental and from 0.269 to 0.613 in task-incremental). Other CL strategies present a smaller margin of improvement. For instance, LwF (from 0.216 to 0.222 in class-incremental, and from 0.262 to 0.349 in task-incremental) and EWC (from 0.190 to 0.220 in class-incremental, and from 0.340 to 0.395 in task-incremental). Shifting the focus on results with EfficientNet (comparing Tables 5, 6), examples include Replay (from 0.417 to 0.655 in class-incremental, and from 0.438 to 0.605 in task-incremental), GenerativeReplay (from 0.313 to 0.517 in class-incremental, and from 0.390 to 0.551 in task-incremental), and Cumulative (from 0.606 to 0.834 in class-incremental, and from 0.596 to 0.656 in task-incremental). Other CL strategies present a more limited but still significant improvement. For instance, CWRStar (from 0.053 to 0.326 in class-incremental, and from 0.171 to 0.367 in task-incremental), and GEM (from 0.218 to 0.312 in class-incremental, and from 0.276 to 0.420 in task-incremental). One counterexample, where the model's performance is higher in the inverse curriculum setting (I2M) is LwF (from 0.203 to 0.165 in class-incremental and from 0.241 to 0.240 in task-incremental). This result shows that different strategies behave differently when presented with a different task ordering. A similar pattern can be observed for ResNet (comparing Tables 7, 8), where the biggest improvement in the class-incremental setting are presented by Cumulative (from 0.614 to 0.788), CWRStar (from 0.102 to 0.331), GenerativeReplay (from 0.313 to 0.447), GDumb (from 0.204 to 0.357), and Replay (from 0.419 to 0.630). Improvements for all other strategies appear minor. Major examples in the task-incremental setting include AGEM (from 0.375 to 0.464), Cumulative (from 0.562 to 0.750), CWRStar (from 0.228 to 0.403), GenerativeReplay (from 0.424 to 0.632), GDumb (from 0.180 to 0.294), Replay (from 0.443 to 0.612). The only exceptions where strategies do not improve in the M2I scenarios are identified with AGEM and EWC in the class-incremental setting. In the task-incremental setting, the exception is MAS.

Another interesting point pertaining to our research question is the opportunity to identify whether learning new tasks favors performance on previously learned tasks, emphasized in our heatmaps. This phenomenon may happen if the model is able to capture similarities between tasks that can be fruitfully leveraged for inference. To show some examples, we focus on task-incremental experiments. In the M2I scenario with VGG9, Fig. 4 shows that different strategies (AGEM, MAS, SI) are able to improve performance on task 1 (Omniglot) after learning task 5 (TinyImageNet). We also observe that multiple strategies (AGEM, EWC, SI, MAS) can leverage the skills learned in task 3 (SVHN) to improve performance on task 1 (MNIST). This result is intuitive since learning the complexity of street numbers in images acquired with a camera strongly benefits the predictive capabilities on an easier dataset from a similar domain, i.e., MNIST. Similar behavior can be observed in Figs. 8 and 12, where AGEM improves the performance on task 2 (Fashion MNIST) after learning task 5 (TinyImagenet). In the I2M scenario for all models: VGG9 (Fig. 5), EfficientNet (Fig. 9) and ResNet (Fig. 13), we can observe that almost all strategies improve the performance on task 3 (Fashion MNIST) after learning task 5 (MNIST). This result shows that the knowledge learned from MNIST can boost the performance on more complex tasks learned before. Overall, results show that task ordering and task similarity can be leveraged to improve performance on a previously learned task, although the currently adopted CL strategies are sparsely able to yield this capability. This consideration paves the way for the design of new strategies that further leverage curriculum task ordering to boost forward and backward transfer (**RQ2**).

An additional consideration pertains to the connection between curriculum learning and the appropriateness of CL metrics in this context. While we are adopting state-of-the-art

metrics that are recognized for properly measuring model accuracy and forgetting in CL scenarios, it should be noted that their interpretation could be counterintuitive in some specific scenarios, where tasks have a varying degree of complexity and metric values are compared with different task orderings. For instance, we note that MNIST is the simplest task and it is presented as the first task (M2I) in the curriculum learning setting. Therefore, it will be considered multiple times in the evaluation protocol, i.e., each time a new task is presented, which may boost the final average result presented by the accuracy metric. In turn, it is more likely for the curriculum learning setting to achieve higher average performance, when a simpler task is presented earlier in the scenario. This behavior poses issues in the interpretability of metric values, which are still unaddressed by currently available metrics. We believe that this aspect should be tackled by future work on metrics for CL.

4.3.1 Summary: RQ2

Overall, results observed across two learning settings (class-incremental, task-incremental) in the two presentation orders (M2I, I2M) show that current methods are not able to fully leverage curriculum learning. One reason may be the fact that most of the CL strategies are heavily impacted by forgetting since they are challenged by the complexities involved in our proposed benchmarks. Comparing the performance and behavior of the CL strategies between M2I and I2M scenarios, we can also observe that different task orderings significantly impact the final outcomes. On the other hand, methods appear to partially benefit from task similarity in some specific cases, as highlighted in our analysis of results. This outcome leads us to the consideration that task similarity could be further exploited by CL strategies to yield models that simultaneously use the knowledge acquired from different tasks to perform better in every single task.

5 Conclusions

In this work, we focused on the problem of benchmarking CL methods, which is often conducted in heterogeneous ways, and with significant simplifications for the learning setting. Specifically, we proposed two novel benchmarks that involve multiple heterogeneous tasks with varying qualities and complexities. Our benchmarks involved six image datasets in increasing (M2I) and decreasing (I2M) difficulty order, following the curriculum learning paradigm. The heterogeneity across datasets allowed us to inject realistic complexities into the learning scenario, resulting in challenging conditions for CL strategies. Particular emphasis was put on the rigorous and reproducible evaluation of model generalization capabilities and forgetting. Our extensive experimental evaluation showed that popular CL strategies, which are known to be robust on commonly adopted scenarios, fail to achieve satisfactory performance with our benchmarks. Moreover, CL strategies are affected by forgetting and are not able to effectively leverage curriculum task ordering to improve their performance and robustness, missing on the opportunity of simultaneously using knowledge from different tasks to perform better in every single task. Our results represent a starting point to assess the impact of curriculum learning on CL strategies. Future work includes the design of new CL strategies that are able to deal with the complexities devised in our benchmarks. Moreover, from an evaluation perspective, new metrics could be investigated to fully capture the spectrum of model behavior with different task orderings. Finally, an interesting line of research pertains to the analysis of the behavior of

non-conventional CL strategies, which are not yet incorporated in known frameworks due to their emerging nature.

Appendix 1: Hyperparameters of included strategies

In our study, as hyperparameters specific to each continual learning strategy, we adopt the configurations reported in their respective research papers. The chosen values are reported in Table 9. For general hyperparameters common to all strategies, we refer the reader to Sect. 4.

Table 9 Hyperparameter values adopted for all continual learning strategies

Strategy	Hyperparameters
EWC	Lambda = 1
	Mode = separate
	Keep importance data = True
LwF	Alpha = 2
	Temperature = 1
MAS	Lambda reg = 1
	Alpha = 0.5
	Mini batch size = 128
SI	Lambda = 1
	Epsilon = 0.001
AGEM	Patterns per experience = 256
	Sample size = 1300
GEM	Memory strength = 0.5
GenerativeReplay	Replay size = 200
	Increasing replay size = False
	Generator strategy = VAETraining
Replay	Replay memory size = 200
CWRStar	cwr layer name = fully connected
PNN	Number of columns = 200
GDumb	Mem size = 200

Author contributions KF: Data Curation, Investigation, Software, Visualization, Writing—DZ: Data Curation, Investigation, Resources, Software, Writing (Review & Editing)—MP, NJ: Resources, Validation, Writing (Review & Editing)—AV, RC: Conceptualization, Supervision, Methodology, Project Administration, Writing

Funding The paper was supported by the Polish Ministry of Science and Higher Education allocated to the AGH UST.

Availability of data and materials The data and materials to reproduce the experiments are available at the following repository URL: https://github.com/lifelonglab/M2I_I2M_benchmark

Code availability The code of the proposed benchmarks is available at the following repository URL: https://github.com/lifelonglab/M2I_I2M_benchmark

Declarations

Conflict of interest All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest in the subject matter or materials discussed in this manuscript.

Ethical approval Not applicable.

Consent to participate This study does not involve human subjects or anysensitive data.

Consent for publication This study does not involve human subjects orany sensitive data.

References

Abel, D., Jinnai, Y., Guo, S. Y., Konidaris, G., & Littman, M. (2018). Policy and value transfer in lifelong reinforcement learning. In *International conference on machine learning* (pp. 20–29). PMLR.

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., & Tuytelaars, T. (2018a). Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 139–154).

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., & Tuytelaars, T. (2018b). Memory aware synapses: Learning what (not) to forget. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision: ECCV 2018* (pp. 144–161). Springer International Publishing.

Ameya Prabhu, P. H. S. T., & Dokania, P. K. (2020). Gdumb: A simple approach that questions our progress in continual learning. Lecture Notes in Computer Science (LNIP) **12347**.

Baker, M. M., New, A., Aguilar-Simon, M., Al-Halah, Z., Arnold, S. M., Ben-Iwhiwhu, E., Brna, A. P., Brooks, E., Brown, R. C., Daniels, Z., et al. (2023). A domain-agnostic approach for characterization of lifelong learning systems. *Neural Networks*

Belouadah, E., Popescu, A., & Kanellos, I. (2021). A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks, 135*, 38–54.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures pp. 437–478.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning* (pp. 41–48).

Cai, Z., Sener, O., & Koltun, V. (2021). Online continual learning with natural distribution shifts: An empirical study with visual data. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8281–8290).

Cano, A., & Krawczyk, B. (2022). Rose: Robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams. *Machine Learning, 111*(7), 2561–2599.

Carta, A., Cossu, A., Hurtado, J., Lomonaco, V., Van de Weijer, J., Hemati, H., et al. (2023). A comprehensive empirical evaluation on online continual learning. arXiv:2308.10328

Chaudhry, A., Ranzato, M., Rohrbach, M., & Elhoseiny, M. (2019). Efficient lifelong learning with a-gem. Salk Institute for Biological Studies arXiv:1812.00420

Corizzo, R., Baron, M., & Japkowicz, N. (2022). Cpdga: Change point driven growing auto-encoder for lifelong anomaly detection. *Knowledge-Based Systems, 247*, 108756.

Cossu, A., Graffieti, G., Pellegrini, L., Maltoni, D., Bacciu, D., Carta, A., & Lomonaco, V. (2022). Is class-incremental enough for continual learning? *Frontiers in Artificial Intelligence*, *5*.

Cun, Y. L. (1998). The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/

David Lopez-Paz, M. R. (2017). Gradient episodic memory for continual learning. arxiv:1706.08840.

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., & Tuytelaars, T. (2021). A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(7), 3366–3385.

De Lange, M., & Tuytelaars, T. (2021). Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 8250–8259).

Díaz-Rodríguez, N., Lomonaco, V., Filliat, D., & Maltoni, D. (2018). Don't forget, there is more than forgetting: new metrics for continual learning. arXiv:1810.13166.

Ebrahimi, S., Meier, F., Calandra, R., Darrell, T., & Rohrbach, M. (2020). Adversarial continual learning. In *European conference on computer vision* (pp. 386–402). Springer.

Faber, K., Corizzo, R., Sniezynski, B., Baron, M., & Japkowicz, N. (2022). Lifewatch: Lifelong wasserstein change point detection. In *2022 International joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Faber, K., Corizzo, R., Sniezynski, B., & Japkowicz, N. (2022). Active lifelong anomaly detection with experience replay. In *2022 IEEE 9th international conference on data science and advanced analytics (DSAA)* (pp. 1–10). IEEE.

Faber, K., Corizzo, R., Sniezynski, B., & Japkowicz, N. (2023). Vlad: Task-agnostic vae-based lifelong anomaly detection. *Neural Networks*.

Gao, K., Wang, H., Cao, Y., & Inoue, K. (2022). Learning from interpretation transition using differentiable logic programming semantics. *Machine Learning* 1–23.

Ghunaim, Y., Bibi, A., Alhamoud, K., Alfarra, M., Al Kader Hammoud, H. A., Prabhu, A., Torr, P. H., & Ghanem, B. (2023). Real-time evaluation in online continual learning: A new hope. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11888–11897).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hihn, H., & Braun, D. A. (2022). Hierarchically structured task-agnostic continual learning. *Machine Learning* 1–32.

Kang, H., Mina, R. J. L., Rizky, S., Madjid, H., Yoon, J., Hasegawa-Johnson, M., Ju-Hwang, S., & Yoo, C. D. (2022). Forget-free continual learning with winning subnetworks. ICML **x**.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., & Hadsell, R. (2016). Overcoming catastrophic forgetting in neural networks. arxiv:1612.00796.

Krawczyk, B. (2021). Tensor decision trees for continual learning from drifting data streams. *Machine Learning, 110*(11–12), 3015–3035.

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

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science, 350*(6266), 1332–1338.

Lange, M. D., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G. G., & Tuytelaars, T. (2022). A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*, 3366–3385.

Le, Y., Yang, X. (1998). Tiny imagenet visual recognition challenge.

Li, Z., & Hoiem, D. (2017). Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(12), 2935–2947.

Lin, Z., Pathak, D., Wang, Y. X., Ramanan, D., & Kong, S. (2022). Continual learning with evolving class ontologies. *Advances in Neural Information Processing Systems, 35*, 7671–7684.

Lin, Z., Shi, J., Pathak, D., & Ramanan, D. (2021). The clear benchmark: Continual learning on real-world imagery. In: Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2).

Lomonaco, V., & Maltoni, D. (2017). Core50: a new dataset and benchmark for continuous object recognition. In: S. Levine, V. Vanhoucke, K. Goldberg (eds.) *Proceedings of the 1st Annual Conference on Robot Learning*, *Proceedings of Machine Learning Research* (vol. 78, pp. 17–26). PMLR. URL https://proceedings.mlr.press/v78/lomonaco17a.html

Lomonaco, V., Maltoni, D., & Pellegrini, L. (2019). Rehearsal-free continual learning over small non-i.i.d. batches. 1st Workshop on Continual Learning in Computer Vision at CVPR2020. https://arxiv.org/abs/1907.03799

Lomonaco, V., Pellegrini, L., Cossu, A., Carta, A., Graffieti, G., Hayes, T. L., De Lange, M., Masana, M., Pomponi, J., Van de Ven, G. M., et al. (2021). Avalanche: an end-to-end library for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3600–3610).

Mallya, A., & Lazebnik, S. (2017). Packnet: Adding multiple tasks to a single network by iterative pruning. arxiv:1711.05769.

Marsocci, V., & Scardapane, S. (2023). Continual barlow twins: continual self-supervised learning for remote sensing semantic segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.
Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. (2011). Reading digits in natural images with unsupervised feature learning.
Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks, 113*, 54–71.
Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., & Wayne, G. (2019). Experience replay for continual learning. *Advances in Neural Information Processing Systems*, *32*.
Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., & Hadsell, R. (2016). Progressive neural networks. arXiv:1606.04671.
Shin, H., Lee, J. K., Kim, J., & Kim, J. (2017). Continual learning with deep generative replay. In: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (eds.) *Advances in Neural Information Processing Systems* (vol. 30). Curran Associates, Inc.
Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.
Song, H., Kim, S., Kim, M., & Lee, J. G. (2020). Ada-boundary: Accelerating dnn training via adaptive boundary batch selection. *Machine Learning, 109*, 1837–1853.
Srinivasan, T., Chang, T. Y., Pinto Alva, L., Chochlakis, G., Rostami, M., & Thomason, J. (2022). Climb: A continual learning benchmark for vision-and-language tasks. *Advances in Neural Information Processing Systems, 35*, 29440–29453.
Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114). PMLR.
Van de Ven, G. M., & Tolias, A. S. (2019). Three scenarios for continual learning. arXiv:1904.07734.
Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747.
Zenke, F., Poole, B., & Ganguli, S. (2017). Continual learning through synaptic intelligence. arxiv:1703.04200.

Authors and Affiliations

Kamil Faber[1] · Dominik Zurek[1] · Marcin Pietron[1] · Nathalie Japkowicz[3] · Antonio Vergari[2] · Roberto Corizzo[3]

✉ Roberto Corizzo
rcorizzo@american.edu

Kamil Faber
kfaber@agh.edu.pl

Dominik Zurek
dzurek@agh.edu.pl

Marcin Pietron
pietron@agh.edu.pl

Nathalie Japkowicz
japkowic@american.edu

Antonio Vergari
avergari@ed.ac.uk

[1] AGH University of Krakow, 30059 Cracow, Poland

[2] University of Edinburgh, Edinburgh EH8 9AB, UK

[3] American University, Washington, DC 20016, USA