Online vs. Offline Continual Learning: a Unified Perspective

Anonymous Authors[1]

Abstract

Continual learning (CL) algorithms are designed to enable neural networks to continually acquire new skills over time without forgetting previously learned ones. There are two main paradigms for CL - offline and online. Offline CL collects and trains on entire task datasets sequentially. In contrast, online CL updates the model more frequently by training incrementally as data arrives, without collecting the entire task dataset. Existing research often reports poorer performance for online CL compared to offline CL, suggesting it faces greater challenges from catastrophic forgetting and underfitting due to its single-pass-through data constraint. In this paper, we challenge this notion by empirically demonstrating that online CL can match or surpass offline CL given equal memory and compute resources. Across three benchmark CL problems with varying data stream sizes and memory budgets, as well as different replay-based CL strategies, our experiments show online CL can outperform its offline counterparts by a margin of 5-10%. We provide some conceptual and theoretical insights into why this *"online replay advantage"* occurs. Specifically, we construct a unified algorithmic framework for online and offline CL showing online CL exhibits a better stability-plasticity tradeoff and a lower bound on generalization error compared to offline CL.

1. Introduction

Deep neural networks have demonstrated remarkable learning capabilities, yet learn in a fundamentally different way than biological brains. The standard deep learning paradigm requires collecting a large dataset upfront to create a stream of IID mini-batches for stochastic gradient descent (SGD) optimization. In contrast, biological networks certainly do not wait to accumulate a lifetime's worth of experience before starting to learn. Instead, human learning begins early in life and continues throughout. We gradually acquire knowledge as we receive new experiences. Continual learning (CL) aims to enable artificial neural networks to continuously acquire skills and knowledge as new data arrives. Two variants of CL have emerged. Offline continual learning formulates the problem as sequentially learning from task datasets, training on a new task only once all the data for this task has been collected before moving to the next. Online continual learning, in comparison, does not collect a specific task dataset but rather trains the model incrementally as data comes in.

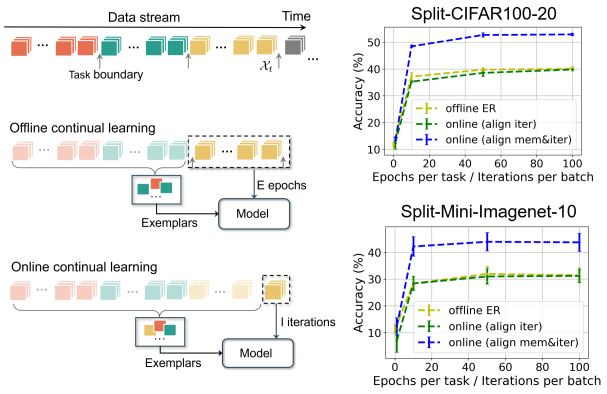


Figure 1. Identifying the *online replay advantage*. Offline CL uses a large space to store task data and typically trains for many epochs per task, while online CL uses a small amount of space to store the current batch and trains for only one or a few iterations per batch. Given equal compute and storage budgets, online CL matches or outperforms offline CL.

A major challenge faced by all continual learning systems, whether online or offline, biological or artificial, is balancing learning new knowledge (i.e., plasticity) while preserving old knowledge (i.e., stability), given limited resources. In particular, when faced with new data, SGD optimization in artificial neural networks easily overwrites previous knowledge, leading to catastrophic forgetting (French, 1999; Delange et al., 2021). To reduce this, techniques have been proposed to increase the stability of online and offline continual learning systems. Replay-based methods, which maintain

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

a subset of past samples (often called exemplars) to regulate gradient optimization, have shown promising results in mitigating forgetting and achieved state-of-the-art CL performance. However, past studies applying these methods in online CL (Mai et al., 2022; Soutif-Cormerais et al., 2023) often report lower performance compared to studies applying the same methods in offline CL (Masana et al., 2022; Buzzega et al., 2020). This suggests online continual learning is inherently more difficult. Specifically, the critical plasticity challenge faced by online CL has been highlighted in (Zhang et al., 2022; Jung et al., 2022): without the ability to store the entire task data and train for multiple epochs, online methods are prone to underfitting. In this paper, we find the plasticity challenge currently observed in online CL primarily arises from memory and computing resources rather than the online setting itself. When compared under equivalent data storage and iteration budgets, online continual learning matches or outperforms offline continual learning across standard CL benchmarks and state-of-the-art replay strategies including experience replay, knowledge distillation, and contrastive replay. We refer to this phenomenon as the *online replay advantage*.

This phenomenon has likely remained unexplored in prior work because online and offline CL are typically studied separately with different experimental setups (see Table 1). For the compute cost, offline CL papers and surveys typically utilize 50-250 training epochs on standard benchmarks, while online CL by design can only use a single epoch. Moreover, while the gradient update steps in online CL can be increased by using repeated iteration, conducting multiple gradient updates for each incoming batch (Zhang et al., 2022; Soutif-Cormerais et al., 2023), the number of iterations is commonly chosen to be less than 10. For the storage cost, a common practice in CL is comparing methods based on a fixed budget of exemplars - a subset of past samples. Less discussed is the sample complexity and storage cost introduced by the new task data, which also occupy the constrained space and are used for the training. At each training session, offline CL needs to store the full task datasets for model training, whereas online CL only stores the most recent incoming batches of the new task.

By accounting for disparities in compute and memory budgets, we provide the first controlled comparison between online and offline CL. Our empirical study reveals two key findings: 1) when online methods use additional training iterations per new batch, the performance gap reduces from 20% to just 5% compared to offline even if the latter is given the standard advantage of extra memory to store task data, and 2) given equivalent iteration and data storage budgets, online CL matches or outperforms offline counterparts by a margin of 5% to 10% across benchmarks, data sizes, and strategies.

Table 1. The compute and data storage cost of online CL (see upper table) and offline CL (see bottom table) on the same CL benchmark (Split-CIFAR100).

Papers	Data Storage (# samples)		Compute
	Exemplars	Working buffer	Iterations
ER	2000-5000	10	1
SCR			1
ER-ACE			1
RAR			10
ER-OBC			1
Survey			3
ICARL	2000-5000	2500-5000	70
EEIL			70
LUCIR			160
DER++			50
MEMO			170
BIC			250
Survey			100

A possible explanation for the online replay advantage is that it yields greater stability, as online CL can allocate more memory to past data given the same storage budget as offline CL. However, we found online approaches do not universally improve stability - this only holds for some strategies like ER, iCaRL, and SCR. For others, like DER++, online CL actually enhances plasticity over offline DER++. This reveals a key difference in the composition of sets of training exemplars between online and offline CL. Offline CL exemplars are sampled solely from past tasks, while online exemplars are sampled from both old and new tasks (see Fig 1). The *new task's exemplars* can be used to drive the plasticity. Overall, online CL appears to more appropriately balance stability and plasticity for a given replay strategy and achieves a better tradeoff than offline learning.

To theoretically understand why online CL leads to a better stability-plastictity tradeoff, we consider online and offline CL in a unified framework where we can vary the relative size of a short-term memory used for storing recent incoming samples vs. the size of a long-term memory used for storing past samples. We derive a generalization bound suggesting online and a new variant that we call "semi-offline CL" achieve lower bounds than offline CL. Additional experiments confirm the performance improvement of online and semi-offline strategies over offline ones.

Our results have several implications: 1) Given sufficient computing resources, online CL accuracy can be increased by performing more training iterations under single-pass-through data constraint. 2) With limited compute and storage, such as embedded devices, it is more effective to train online as data arrives than to collect full task datasets. 3) The relative sizes of working and episodic memory significantly influence the stability-plasticity tradeoff. A performance

boost can be achieved for a CL algorithm by adjusting this ratio. 4) Online and offline continual learning are not fundamentally distinct paradigms: unified algorithms can be developed incorporating insights from both.

2. Related work

Online and offline continual learning. General continual learning (Delange et al., 2021; Buzzega et al., 2020) is an idealized scheme for learning from an infinite data stream, with desiderata like constant memory, online learning, no task boundaries, no task labels, and graceful forgetting. Various relaxations exist with different assumptions. Early work focused on task-incremental settings (Mallya & Lazebnik, 2018; Serra et al., 2018) that assume access to task labels during training/testing. Despite promising results, relying on a task oracle is impractical. Recent class-incremental and domain-incremental learning approaches remove this assumption (Mirza et al., 2022; Masana et al., 2022; Van de Ven & Tolias, 2019). Nevertheless, these methods still require the knowledge of task boundaries to allow multi-epoch training over tasks. The online continual learning paradigm (Chaudhry et al., 2019; Aljundi et al., 2019; Mai et al., 2022) eliminates task boundary assumptions, performing single-pass learning over streams.

Forgetting mitigation techniques. Continual learning algorithms address catastrophic forgetting in three main ways: replay-based methods (Chaudhry et al., 2019; Aljundi et al., 2019) store and replay past samples to mitigate forgetting; regularization-based methods (Rebuffi et al., 2017; Li & Hoiem, 2017) use regularization losses to encourage retention of past knowledge; architecture-based methods (Mallya & Lazebnik, 2018; Serra et al., 2018) separate parameters for different tasks to avoid interference.

Memory and compute. Memory and compute resources are crucial in continual learning. If there is unlimited memory and compute power, the continual learning problem can be solved by retraining the model from scratch on the cumulative data whenever new data arrives. Therefore, a common assumption in continual learning research is that only a subset of the previous data can be stored for continually updating the model, rather than all of it. The effectiveness of different continual learning methods is often compared by giving a fixed exemplar budget, which refers to the number of past data samples (exemplars) stored. In addition to the number of exemplars, some recent works also consider other aspects, such as the storage cost per sample, with one work proposing the use of data compression to store more data at a lower quality (Wang et al., 2021), and the storage cost of models with one work comparing CL methods under a joint storage budget of exemplars and models. However, these works often focus on offline continual learning and do not take into account the storage cost of new task data. In contrast, our work jointly considers the cost of new and old task data in memory management in continual learning. This presents an orthogonal consideration to other factors such as storage cost per sample or models. We use data storage budget or memory budget interchangeable in this paper to refer the number of samples from the new and past task stored for CL.

In terms of compute, the training epochs in offline CL are often considered hyperparameters and not aligned when comparing different CL techniques. A recent study (Prabhu et al., 2023) compares various CL techniques with a limited iteration budget and concludes that rehearsal appears to be the most effective in this scenario. However, this study assumes no memory constraints with all the data stream samples stored in the memory. In contrast, our work focuses on unbounded memory with an aligned compute budget.

A recent work (Ma et al., 2023) considers the power consumption of CL in hardware implementation. Instead of employing a single memory structure with the same IO speed, it investigates the use of a hierarchal memory structure with a small memory with fast access and large storage with slow access to achieve a better tradeoff between accuracy and energy. In comparison, our study emphasizes on accuracy and stability-plasticity tradeoff of CL. We assume a single-level memory structure and set the compute budget based on iteration per batch.

3. Problem setting

Online and offline continual learning have been studied as separate research areas. We first formalize the problem setting and terminology to describe the two paradigms.

Data stream. Given a non-stationary (potentially infinite) stream of data $\mathcal{D}_t = \cup_t \mathcal{X}_t$: at each time step t, a continual learning algorithm $\mathcal{A}$ receives an incoming batch of data samples $\mathcal{X}_t = \{\mathbf{x}_i, y_i\}_{i=1,..,|\mathcal{X}_t|}$ that are drawn from the current data distribution $\mathbb{P}_t(x, y)$. The goal is to minimize the empirical risk on all the data seen so far:

$$\min_{\theta} \mathcal{R}(\theta) = \min_{\theta} L(\cup_t \mathcal{X}_t; \theta). \tag{1}$$

with a loss function L, a CL network function $f : x \to y$, and its associated parameters θ.

Non-stationary distribution and task boundaries. Consider a non-stationary data stream, where the distribution of batch data $\mathbb{P}_t(x, y)$ may change at any time. The period where the data distribution $\mathbb{P}_t(x, y)$ stays the same is often called a *task* or *experience*, and a time step is a task boundary if distribution changes occur, i.e., $\{T_i\} \doteq [t|\mathbb{P}_t(x, y) \neq \mathbb{P}_{t-1}(x, y)]$. Given the task boundary of a data stream, each task data can be denoted as $\mathcal{C}_i = \cup_{t\in[T_i,T_{i+1}]} \mathcal{X}_t$ and $\cup_i C_i = \mathcal{D}_t$.

While online CL operates on data stream batches $\mathcal{X}_t$ directly, offline continual learning requires the additional knowledge of task boundaries to collect the task dataset (Delange et al., 2021).

Exemplars. Given a bounded exemplar memory $\mathcal{M}$, a stream of data $\mathcal{D}_t = \cup_t \mathcal{X}_t$ ($|M| < |D|$), and a sample selection policy π: at each time step t, an online exemplar management algorithm takes parts of the incoming batch into the memory and ejects some of the previous data $\mathcal{M}_t \subset_\pi \mathcal{M}_{t-1} \cup \mathcal{X}_t$; for each task i, an offline exemplar management algorithm takes parts of the task data into the memory and ejects some of the previous data $\mathcal{M}_i \subset_\pi \mathcal{M}_{i-1} \cup \mathcal{C}_i$.

In this work, we focus on reservoir sampling (Vitter, 1985), a simple yet effective exemplar management strategy that randomly selects a sample of k items from a larger population of unknown or very large size, in an unbiased manner.

Offline continual learning $\mathcal{A}(\mathcal{C}_i, \mathcal{M}_{i-1}, \theta_{i-1})$. Given a data stream with task boundaries, $\mathcal{D}_t = \cup_i \mathcal{C}_i$, an exemplar set $\mathcal{M}$, and a loss function L_θ, the model is updated by going over the task $\mathcal{C}_i$ for E epochs:

$$\theta \leftarrow \theta - \eta \nabla L(\mathcal{X}_{\mathcal{C}_i} \cup \mathcal{X}_M; \theta), \text{where } \mathcal{X}_{\mathcal{C}_i} \sim \mathcal{C}_i, \mathcal{X}_M \sim \mathcal{M}. \tag{2}$$

Online continual learning $\mathcal{A}(\mathcal{X}_t, \mathcal{M}_{t-1}, \theta_{t-1})$. Given a data stream $\mathcal{D}_t = \cup_t \mathcal{X}_t$, an exemplar set $\mathcal{M}$, and a loss function L_θ, the model is updated by computing gradients using the same batch $\mathcal{X}_t$ for I iterations:

$$\theta \leftarrow \theta - \eta \nabla L(\mathcal{X}_t \cup \mathcal{X}_M; \theta), \text{ where } \mathcal{X}_M \sim \mathcal{M}. \tag{3}$$

The most common choice in online CL is using a single iteration $I = 1$. However, recent work (Zhang et al., 2022) shows that although simply increasing iterations can lead to worse performance, combining augmentation with repeated iteration (i.e., repeated augmented replay) consistently improves online CL performance. Thus, in our experiments, augmentation is employed in all cases.

4. Main experiments

4.1. Experiment setup

Memory and Compute. We construct a structured comparison between online and offline CL by taking into account memory and compute cost. We present experimental results with two scenarios: 1) *Aligned compute and exemplar memory*. The number of epochs in offline CL is equal to the number of iterations in online CL, and the size of the exemplar memory is the same, i.e., $E = I$ and $|M_{online}| = |M_{offline}|$. In this setting, online CL leads to smaller data storage cost than offline CL. 2) *Aligned compute and data storage*. The number of epochs in offline CL is equal to the number of iterations in online CL, and the total storage is the same, i.e., $E = I$ and $|M_{online}| + |\mathcal{X}_t| = |\mathcal{C}_i| + |M_{offline}|$. In this setting, online CL stores more exemplars than offline CL.

The experiments involve three standard CL benchmarks: Split-CIFAR100 with 20 tasks, Split-Mini-ImageNet with 10 tasks, and CORE50 with 9 tasks. The task size in these three benchmarks is 2500, 5000, and 12000 respectively. We consider the data stream batch size to be 50. In the main experiment, we set the exemplar budget for offline CL approaches to 2k. In ablation studies with other exemplar budgets, we also consider sizes of 1k, 5k, and 10k. Under the aligned compute setting, the exemplar budget for online CL is equal to that of offline CL. In the second setting, where compute and storage are both aligned, the online exemplar budget is computed as $M_{offline} + C - B$ where C and B are task size and batch size. In the main experiment, we evaluate four compute budgets (specifying the offline epoch count and online iteration count respectively): 1, 10, 50, and 100. We use ResNet-18 for all experiments. We employ standard data augmentation (random cropping and flipping) for most experiments, except for SCR and CORE50, which utilize cropping, flipping, color jittering, and grayscale. Training details can be found in the Appendix C.

Strategies. We consider three types of replay-based approaches: 1) *Direct rehearsal*. Our main experiment is focused on experience replay (Chaudhry et al., 2019), which is a simple approach that achieves competitive performance especially in large-scale settings (Prabhu et al., 2023). ER incorporates past exemplars directly in the training via cross-entropy loss. 2) *Knowledge distillation*. Many replay-based methods leverage knowledge distillation to construct a regularization loss, using a past model as the teacher and the current model as the student (Li & Hoiem, 2017; Rebuffi et al., 2017; Buzzega et al., 2020; Hinton et al., 2015). A classic method is iCaRL which maintains a past model and computes distillation loss based on the past network's outputs related to old classes. Instead of computing logits based on a past model, the distillation loss of DER++ uses the network's logits sampled throughout the optimization trajectory, and the distillation loss is computed over past exemplars. 3) *Contrastive replay*. Some recent works (Cha et al., 2021; Mai et al., 2021; Khosla et al., 2020) investigate the use of self-supervised learning techniques to learn a strong representation to reduce forgetting. A representative method is SCR (Mai et al., 2021), which replaces cross-entropy with contrastive loss to capture more information about exemplars and achieves state-of-the-art performance.

Metrics. CL performance is measured by the final accuracy after training on all tasks, defined as

$$A_T = \frac{1}{T} \sum_{j=1}^{j=T} a_{T,j} \tag{4}$$

where $a_{i,j}$ denotes the model's accuracy on the held-out

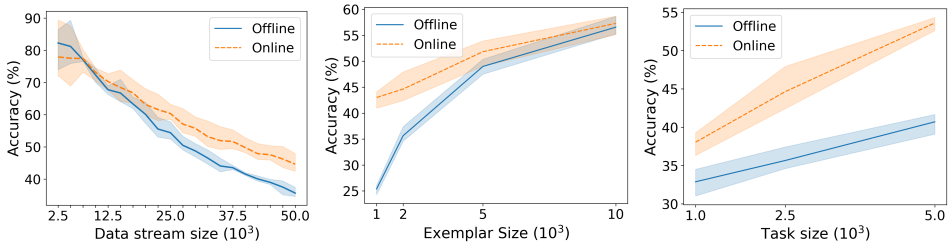


Figure 2. The *online replay advantage* in different problem settings becomes larger with a longer data stream sequence N, a smaller exemplar budget M, and a larger task data size C. This experiment employs ER in various Split-CIFAR100 formulations with different storage budgets and task sizes.

Table 2. Comparing online and offline continual learning approaches with three popular CL datasets under two settings: 1) aligned compute and exemplar budget, 2) aligned compute and storage budgets. The compute budget is 50 iterations/epochs. Exemplar budget in setting 1 is 2k and the total storage budget in setting 2 for each dataset is task size + 2k.

	WORKING BUFFER		ER	ICARL	DER++	SCR
S-CIFAR100-20	2500	OFFLINE	35.1 ± 0.8	44.8 ± 1.1	47.2 ± 0.4	45.1 ± 0.4
	50	ONLINE (1)	35.0 ± 1.2	42.7 ± 1.9	40.4 ± 2.0	43.7 ± 0.3
	50	ONLINE (2)	44.6 ± 2.4	50.0 ± 1.6	50.9 ± 0.9	51.9 ± 0.5
S-MINI-IMAGENET-10	5000	OFFLINE	31.2 ± 1.1	42.5 ± 1.2	41.0 ± 2.7	46.3 ± 0.4
	50	ONLINE (1)	29.1 ± 1.3	36.1 ± 0.6	33.6 ± 0.6	42.5 ± 0.3
	50	ONLINE (2)	43.7 ± 1.2	46.8 ± 1.5	46.4 ± 1.3	51.8 ± 0.7
S-CORE-9	12000	OFFLINE	40.1 ± 2.4	45.8 ± 1.6	38.2 ± 1.5	62.1 ± 1.3
	64	ONLINE (1)	39.1 ± 3.5	47.5 ± 2.4	41.4 ± 2.2	59.3 ± 1.5
	64	ONLINE (2)	50.7 ± 1.7	50.1 ± 1.6	46.7 ± 2.6	69.7 ± 0.3

test set of task j after training on task i. Other metrics are "forgetting" (Chaudhry et al., 2018), which is defined as $F_T = -\frac{1}{T-1}\sum_{i=1}^{T-1}(a_{T,i} - \max_{l\in 1\ldots T-1} a_{l,i})$ and the related metric "backward transfer" (Lopez-Paz & Ranzato, 2017): $B_T = \frac{1}{T-1}\sum_{i=1}^{T-1} a_{T,i} - a_{i,i}$. Stability and plasticity are defined as: $\frac{1}{T}\Sigma_{i=1}^{T} a_{i,i}$ and $\frac{T-1}{T}B_T$ respectively (Zhang et al., 2022). Under these definitions, A_T = stability + plasticity

4.2. Main findings

The effect of increasing iterations. Despite the current practice of offline and online continual learning (CL) approaches adopting different compute setups (100-200 epochs per task vs. 1-10 iterations per batch), we observe that increasing compute influences online and offline CL similarly (see Fig 1 (b)). More specifically, when increasing training iterations from 1 to 10, there is a rapid increase in accuracy in online CL, corroborating the findings reported in (Zhang et al., 2022). However, it is noteworthy that while their experiment utilizes a complex augmentation strategy (RandAugment), our results demonstrate that this finding persists even with simple augmentation techniques such as cropping and flipping. More importantly, although previous online CL practices adopt iterations less than 10, we find that increasing the iterations from 10 to 50 still produces a significant performance boost of around 5% for four CL algorithms (see Figure 6 in the Appendix for the results of iCARL, DER, and SCR). Further increasing iterations from 50 to 100 does not significantly increase or degrade performance.

Aligned compute and exemplar budget (setting 1). When comparing online and offline CL with aligned exemplar budgets, we observe that increasing the number of iterations beyond a single iteration brings the performance of online CL closer to its offline counterparts. However, the gap between the two does not vanish in most cases, with offline CL still outperforming online (1) in Table 2 in most cases.

It is worth noting that even when compute and exemplar budgets are aligned, several key distinctions persist between online and offline CL approaches regarding the frequency of model updates and exemplar management (see Algorithm 2 of Appendix B for the pseudo-codes of online and offline ER). More specifically, given an iteration budget of K, in offline CL, the incoming gradient is computed using data batches sampled from the entire task dataset, whereas online CL has to reuse the *same incoming batch* for K consecutive gradient steps (multi-step SGD), potentially hindering the acquisition of new knowledge (plasticity). On the other hand, given the same exemplar budget, online CL exemplars need to summarize a broader range of data encompassing

previous tasks and *past samples of the current task*, whereas offline CL exemplars only need to cover previous tasks. Consequently, online CL faces an inherent disadvantage compared to offline CL under setting 1. Based on Table 2, we find that the performance gap appears to be smaller for direct rehearsal strategies like ER and SCR. Larger gaps are observed in knowledge-distillation-based methods such as iCARL and DER++, suggesting that these methods may be more sensitive to the noisy multi-step optimization, thus rendering them relatively less effective in online settings.

Aligned compute and storage (setting 2). Given a total data storage budget and equivalent compute, online CL still faces the challenge of noisy mulitple-step SGD but enjoys a larger exemplar space than offline CL. We investigate how the interplay of these two factors affects the performance. We observe the online ER substantially outperforms offline ER (see Fig 1 (b)). In fact, online ER with 10 iterations already outperforms offline CL with 200 iterations. We referred as the phenomenon of superior performance of online CL under setting 2 as *online replay advantage*. One implication of this finding in applications with limited storage space like embedding devices, performing online CL enjoyes several advantages over offline CL including better CL accuracy, fewer iterations required, and no requirement on the knowledge of task boundary.

Different replay techniques make use of exemplars to preserve past knowledge in different ways. A replay method with a very strong forgetting mitigation design may not need a large number of exemplars from previous tasks to perform well. Thus, one interesting question is how the online advantage gap changes with different forgetting designs. Table 2 presents the results of different CL strategies (ER, iCaRL, DER++ and SCR) on three standard CL benchmarks, showing the online replay advantage to persist in all these experiments, although the online advantage gap in knowledge distillation methods (iCaRL and DER++) and the contrastive replay method SCR seems to be generally smaller than for ER.

Continual learning problem structure. We further investigate how the online replay advantage is influenced by different problem settings (see Fig 2). First, we investigate the influence of data stream sizes by showing the performance at different stages of CL problems. It seems that with longer task sequences the online advantage becomes more obvious. Second, we consider different sizes of total data storage. The online advantage is observed in all cases but the advantage decreases for larger data storage budgets. Third, we study the effect of task size, which is determined by the frequency of distribution changes in the data stream. Problems with different task sizes are constructed by splitting the CIFAR100 dataset into 10, 20, or 50 tasks. With smaller tasks (50 tasks), online and offline ER achieve similar performance, whereas the online advantage gap seems to increase for larger task sizes. In summary, we observe that online CL consistently matches or outperforms offline CL. The advantage gap of online ER becomes more pronounced for smaller budgets, longer data streams, and larger task sizes. Interestingly, the performance gap appears to be also correlated with the overall difficulty of the continual learning problem and accuracy. In more challenging continual learning scenarios that exhibit lower accuracy, online (2) tends to outperform offline by a larger margin.

Table 3. Comparing online and offline CL in task-incremental (TI), domain-incremental (DI) and pretrained class-incremental settings.

CL Settings	TI	DI	Pretrained-CI
Dataset	CIFAR100	CLRS	Mini-ImageNet
Offline ER	83.4 ± 1.3	34.2 ± 2.6	36.6 ± 0.9
Online (1)	79.7 ± 1.4	34.3 ± 0.7	36.2 ± 1.3
Online (2)	83.1 ± 2.3	36.8 ± 1.3	48.3 ± 1.1

4.3. Different CL settings

Previous experiments focused on class-incremental setting and training the model from scratch. We further investigate this phenomenon in task-increment, domain-incremental and pretrained settings (see results in Table 3).

Pre-trained Setting. When initializing the model with a pre-trained ResNet18, the performance of both online and offline Experience Replay is significantly improved (see Figure 10 in the Appendix). However, we observe a similar trend in comparing online and offline CL, with offline ER slightly outperforming online (1) and significantly underperforming online (2).

Domain-Incremental Setting. We also observe the *online replay advantage* in a domain-incremental dataset, CLRS (Li et al., 2020; Zhang et al., 2022). However, the performance gap between online (2) and offline seems to be smaller (around 2%) than in the class-incremental case (10%). This result suggests that in problems with similar tasks, the online advantage gap may become smaller. Our theoretical analysis (Corollary 1) in Section 5.3 also indicates that the advantage gap is affected by task similarity.

Task-Incremental Setting. Interestingly, online (2) and offline CL achieve very similar performance in our task-incremental experiment, which employs ER on CIFAR100 with 50 iterations. A key difference between the class-incremental and task-incremental settings is that the latter employs a parameter-isolation mechanism in the multi-head classifier. As suggested in (Ye & Bors, 2022), parameter isolation may reduce the discrepancy distance (Mansour et al., 2009) between different domains. And our theoretical result in 5.3 suggests that online advantage gap is influenced by

the discrepancy distance between domains.

5. Unified online and offline continual learning

To further investigate why online CL is able to achieve a better stability-plasticity tradeoff than offline CL, we build a unified framework for online and offline CL. This framework also enables us to study whether there is a "sweet spot" between online and offline continual learning that optimizes the stability-plasticity tradeoff.

Algorithm 1: Unified Continual Learning

function Training($K, \theta, \mathcal{M}_{short}, \mathcal{M}_{long}$)
  for epoch = 1,...,K **do**
    for $\mathcal{X}_{short}$ in $\mathcal{M}_{short}$ **do**
      $\mathcal{X}_{long} \sim M_{long}$,
      $\theta \leftarrow \theta - \eta \nabla L(\mathcal{X}_{short} \cup \mathcal{X}_{long}; \theta)$
  return θ
function UCL($\mathcal{X}_t, \theta, \mathcal{M}_{short}, \mathcal{M}_{long}$,)
  $\mathcal{M}_{short} \leftarrow \mathcal{M}_{short} \cup \mathcal{X}_t$
  if $\mathcal{M}_{short}$ is full **then**
    $\theta \leftarrow$ Training($K, \theta, \mathcal{M}_{short}, \mathcal{M}_{long}$)
    $\mathcal{M}_{long} \subset_\pi \mathcal{M}_{short} \cup \mathcal{M}_{long}$
    $\mathcal{M}_{short} \leftarrow \emptyset$
  return $\theta, \mathcal{M}_{short}, \mathcal{M}_{long}$

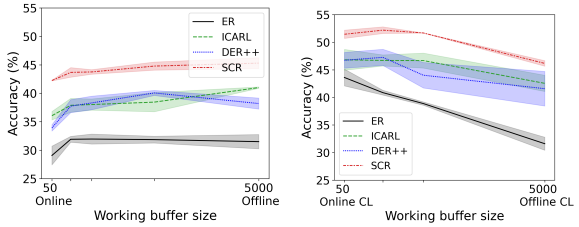
5.1. A unified CL framework

Unified continual learning $UCL(M_s, M, K)$. Given a compute budget of K, a total data storage budget M, a stream of data $\mathcal{D}_t = \cup_t \mathcal{X}_t$ ($M < |\mathcal{D}_t|$) with a batch size of B and sample selection policy π, the storage space is allocated into two memories: a short-term memory of size M_s ($M_s > B$) and a long-term memory of size $M - M_s$. The short-term memory $\mathcal{M}_{\text{short}}$ greedily stores recent batches from the data stream until it is full: $\mathcal{M}_{\text{short}} = \cup_{t-n+1,\ldots,t} \mathcal{X}_t, n = \frac{M_s}{B}$. Each model training session begins when $\mathcal{M}_{\text{short}}$ is full and includes $K \times n$ gradient steps following Equation 5.

$$\theta \leftarrow \theta - \eta \nabla L(\mathcal{X}_{long} \cup \mathcal{X}_{short}; \theta), \quad \text{where } \mathcal{X}_{long} \sim \mathcal{M}_{long},\ \mathcal{X}_{short} \sim \mathcal{M}_{short}. \tag{5}$$

After the model training session, $\mathcal{M}_{\text{short}}$ is emptied, with some of the short-term memory samples moved into $\mathcal{M}_{\text{long}}$ based on some sample selection policy π: $\mathcal{M}_{long} \subset_\pi \mathcal{M}_{short} \cup \mathcal{M}_{long}$.

The training procedure of unified continual learning is shown in Algorithm 1. To ensure the current new batch can be fitted into the short-term memory and used for model training and sample selection, an assumption of UCL is $M_s \geq B$. Online and offline CL both follow this UCL framework. The only difference between the two is the size of the working buffer: when the short-term memory is as large as the task size C, $UCL(C, M, K)$ is exactly offline CL with K epochs per task. When the short-term memory is as large as the batch size B, $UCL(B, M, K)$ is online CL with K iterations per batch. When the task size is equal to the batch size, $M_s = B = C$, and online and offline CL are the same.



(a) Setting 1 (b) Setting 2

Figure 3. Semi-offline results: (a) aligned compute and exemplar budget: fixing exemplar budget, increasing working buffer size improves CL performance. (b) aligned compute and storage budget: given a total storage budget, increasing the working buffer size decreases CL performance (Experiments in Split-Mini-ImageNet).

Semi-offline CL. Apart from the two extreme cases of online and offline continual learning, this unified framework unveils an unexplored spectrum of working buffer sizes that lie in between these extremes. As shown in Algorithm 1, the model is updated every M_s incoming samples, where M_s is the working buffer size. Therefore, different working buffer sizes essentially correspond to different model update frequencies. We refer to these settings with a working buffer size $B < M_s < C$, where B and C are the batch size and the task size, respectively, as *semi-offline continual learning*.

5.2. Empirical results: How frequently to update model

The unified CL framework allows us to explore the important question of how often we should update the model. Due to the cost of deploying models, many companies cannot update them for every new data batch but instead choose to update them based on collected micro-batches (i.e., short-term buffers) daily, weekly, or monthly (Huyen, 2022). The frequency of model updates is a topic often discussed in online and streaming learning to assess its impact on fast adaptation. However, it remains under-explored in continual learning to understand its effect on the model's information retention within the context of continual learning.

Semi-offline in setting 1: the effect of the working buffer. Under a fixed exemplar memory budget, Fig 3 (a) shows that increasing the working buffer size leads to better CL performance. This result suggests that accumulating larger

batches of data before updating the model is beneficial for overall CL performance, given a fixed exemplar budget. One possible explanation is that a larger working buffer mitigates the noise introduced by the multiple-step SGD and may lead to better acquisition of knowledge from the new tasks. To investigate how well the model adapts to the new tasks, we report the next-batch accuracy following (Cai et al., 2021; Ghunaim et al., 2023), which is also known as the interleaved-test-then-train metric (Montiel et al., 2018) in the streaming learning literature. Fig 4 indicates that larger working buffers indeed yield better final performance on the new tasks. However, a notable downside of increasing the working buffer size is the delayed responsiveness of the system to new data due to the less frequent model updates shown in Fig 4.

Semi-offline in setting 2: the trade-off between working buffer and exemplar buffer. The results above show increasing working buffer leads to better CL performance. This raises an interesting question of how to trade off the working buffer and exemplar buffer sizes given a total storage budget. We investigate the effectiveness of semi-offline approaches under the same data storage and compute budget. Fig. 3 (b) present the semi-offline results with Split-Mini-Imagenet. A similar result for Split-CIFAR100 can be found in the appendix fig 8. Generally, we observe the performance of CL seems to gradually degrade as the working buffer size increases (Fig. 3 (b)). This performance degradation seems to be smaller in some replay methods than in others (e.g., ER). In all approaches, online and semi-offline are significantly better than offline CL. A few semi-offline cases (with a small working buffer) seem slightly better than online CL but this difference is not statically significant. Overall, instead of observing a clear sweet spot between online and offline CL, the result seems to show online CL is a competitive strategy along the online-offline continuum.

Stability and plasticity analysis. In terms of stability and plasticity, we find that reducing the working buffer size leads to various types of stability-plasticity behaviors. As shown in Fig. 5 (c), for some replay methods (ER, iCaRL and SCR), as the working buffer size decreases, stability seems to increase at the cost of decreasing plasticity. For DER++, reducing the working buffer to different extents leads to different stability-plasticity dynamics such as higher plasticity with lower stability, higher plasticity and higher stability, lower plasticity, and lower stability. Interestingly, regardless of these various stability-plasticity changes, reducing the working buffer consistently leads to a better stability-plasticity tradeoff overall. To establish a conceptual understanding of the stability and plasticity changes, we consider the distribution composition of long-term buffers. While in offline CL, the long-term buffer consists of exemplars from previous tasks, in online and semi-offline CL, the long-term buffer contains a mix of exemplars from both previous tasks and the current task, where the ratio is based on reservoir sampling. Specifically, given a data stream of size N, task size C, and short-term memory size M_s, the ratio for new task exemplars is $p = \frac{C-M_s}{N-M_s}$, which increases with a smaller M_s. Hence, as the working buffer decreases, more space is used for saving past tasks'exemplars, which drives stability, but these exemplars are mixed with an increasing ratio of new task exemplars, which drives plasticity.

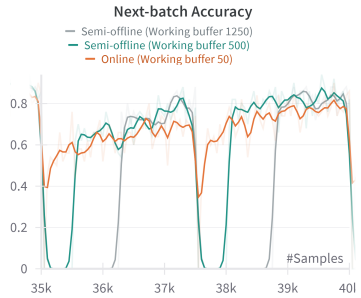


Figure 4. Next-batch evaluation of online and semi-offline CL under setting 1 (aligned compute and exemplars). Semi-offline ER achieves better adaptation quality, resulting in higher final accuracy on the new task, while online ER achieves faster adaptation. As an example, we illustrate the learning progression from Task 14 to 16 in Split-CIFAR100 here. Full results are available in Appendix Figure 11. Gaussian smoothing has been applied to the curves to enhance readability.

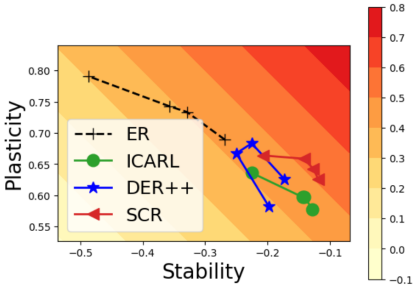


Figure 5. Stability and plasticity analysis of a continuum of online, semi-offline and offline CL in setting 2 (aligned compute and storage).

5.3. Theoretical analysis

To achieve a deeper understanding of this phenomenon, we analyze the generalization bound of the unified continual learning framework.

Traditional generalization bounds concern the setting where the training and the test dataset are from the same distribution. To analyze the generalization ability in non-stationary situations, some transfer learning research derives generalization bound based on the *discrepancy distance* (see Defi-

nition 1) between different distribution domains (Mansour et al., 2009). This kind of generalization bound predicts how the model trained in one domain (i.e., the source domain) will perform in another domain (i.e., the target domain).

Definition 1 (Discrepancy distance) (Mansour et al., 2009). Let H be a set of functions mapping X to Y and let $L : Y \times Y \to R^+$ define a loss function over Y. The discrepancy distance between two distributions Q_1 and Q_2 over X is defined by

$$\mathrm{disc}_L\left(Q_1, Q_2\right) \doteq \max_{h,h' \in H} \left|\mathcal{L}_{Q_1}\left(h', h\right) - \mathcal{L}_{Q_2}\left(h', h\right)\right|.$$

where the expected loss of two functions over a distribution is denoted as $\mathcal{L}_{\mathbb{Q}}(f, g) \doteq \mathrm{E}_{x\sim\mathbb{Q}}[L(f(x), g(x))]$.

Since continual learning concerns the generalization ability on all seen task domains, recent work (Ye & Bors, 2022) regards all seen tasks as the target domain and derives a generalization bound for online CL as follows. Let $\mathbb{D}$ and $\mathbb{M}$ denote the expected probability distributions of the data stream and the stored samples respectively. Let $\hat{\mathbb{M}}$ denote the empirical distribution of stored samples with a finite sample size of M. The true labeling function is defined as h_y. Given the optimal solutions $h^*_{\mathbb{M}} \doteq \mathrm{argmin}_{h\in H}\, \mathcal{L}_{\mathbb{M}}\left(h, h_y\right)$ and $h^*_{\mathbb{D}} \doteq \mathrm{argmin}_{h\in H}\, \mathcal{L}_{\mathbb{D}}\left(h, h_y\right)$, the generalization bound is presented in Theorem 1 (Ye & Bors, 2022).

Theorem 1 (Ye & Bors, 2022) . Let H be a hypothesis set bounded by some $A_0 > 0$ for the loss function $L : L(h, h') \leq A_0$, for all $h, h' \in H$. Assume that the loss function L is symmetric and obeys the triangle inequality. Then, for any $h \in H$ and any $\delta > 0$, with probability at least $1 - \delta$, the following generalization bound holds:

$$\begin{aligned}\mathcal{L}_{\mathbb{D}}(h, h_y) \leq \mathcal{L}_{\hat{\mathbb{M}}}(h, h^*_{\mathbb{M}}) + \widehat{\Re}_{\mathcal{M}}(H) + 3A_0\sqrt{\frac{\log\frac{2}{\delta}}{2M}} \\ + \mathrm{disc}_L(\mathbb{D}, \mathbb{M}) + \mathcal{L}_{\mathbb{M}}(h^*_{\mathbb{M}}, h^*_{\mathbb{D}}) + \mathcal{L}_{\mathbb{D}}(h^*_{\mathbb{D}}, h_y),\end{aligned} \tag{6}$$

where $\widehat{\Re}_{\mathcal{M}}(H)$ is the empirical Rademacher complexity of the hypothesis set H over a sample set $\mathcal{M}$.

Theorem 1 reveals the relationship between generalization and the discrepancy distance $\mathrm{disc}_L(\mathbb{D}, \mathbb{M})$ between the true data stream distribution $\mathbb{D}$ and the expected distribution $\mathbb{M}$ of the stored memory samples. Crucially, minimizing discrepancy distance $\mathrm{disc}_L(\mathbb{D}, \mathbb{M})$ leads to a lower generalization bound.

However, one limitation of (Ye & Bors, 2022) is that this analysis is focused on online CL, and when investigating the source domain, only the exemplar buffer is considered as the source domain: although the incoming batch samples are also used for training, they are overlooked in generalization-bound analysis. To address this problem, we reexamine the finding of Theorem 1 in the online, semi-offline, and offline CL cases, by taking into account both memory buffer and working buffer as the source domain. More specifically, we analyze the discrepancy distance $\mathrm{disc}_L(\mathbb{D}, \mathbb{M})$ in the unified CL framework. Our main result is shown in Proposition 1.

Proposition 1. Assume $\mathbb{P}_+$ denotes the probability distribution of the most recent task $\mathcal{C}_i$ and $\mathbb{P}_-$ denotes the probability distribution of all past tasks $\cup_{1,\ldots,i-1}\mathcal{C}$. Given the number of samples seen in the data stream $N = \sum_t |\mathcal{X}_t|$ and the number of samples seen in the previous tasks $N^- = \sum_{k=1}^{k=i-1} |\mathcal{C}_k|$, we have:

$$\mathrm{disc}_L(\mathbb{D}, \mathbb{M}) = \frac{N^- M_s(N - M)}{NM(N - M_s)}\,\mathrm{disc}_L(\mathbb{P}_-, \mathbb{P}_+). \tag{7}$$

Proof. See Appendix A.1. □

Proposition 1 reveals the interplay between data storage allocation, task similarity, and generalization capability in continual learning. In the IID setting, where $\mathbb{P}_- = \mathbb{P}_+$, we have $\mathrm{disc}_L(\mathbb{D}, \mathbb{M}) = 0$, leading to the traditional generalization bound. When tasks are dissimilar and the loss cannot trivially minimize discrepancies, $\mathrm{disc}_L(\mathbb{D}, \mathbb{M}) \neq 0$, and the storage mechanism has a non-trivial impact on the generalization bound.

Online-offline gap. Given a total data storage budget, the generalization bound is affected by the discrepancy distance. We hypothesize the online replay advantage gap observed under equivalent storage budgets arises from the differences in the discrepancy distance between online and offline CL, i.e., $R_L(N, M, C) \doteq \mathrm{disc}_L(\mathbb{D}, \mathbb{M}_{offline}) - \mathrm{disc}_L(\mathbb{D}, \mathbb{M}_{online})$.

Corollary 1. Given a data steam size N with a batch size of B, a memory budget M, and the size of the most recent task C, the online-offline gap is:

$$R_L(N, M, C) = \frac{C - B}{N - B} \times \frac{N - M}{M}\,\mathrm{disc}_L(\mathbb{P}^-, \mathbb{P}^+). \tag{8}$$

Proof. The proof is based on the short-term memory sizes in online and offline CL are batch size and task size respectively. See Appendix A.2 for more detail. □

Problem structure. Corollary 1 reveals three key insights into how the online-offline gap is affected by the problem structure: 1) The online advantage gap increases as more data arrives ($\partial R_L/\partial N > 0$); 2) The gap diminishes with larger memory ($\partial R_L/\partial M < 0$), converging as $M \to N$; 3) Online rehearsal becomes more crucial as task size grows (see Fig 1 (b)) ($\partial R_L/\partial C > 0$). This finding aligns with the empirical results in Fig 2.

Forgetting mitigation strategies. Corollary 1 also characterizes how the discrepancy distance gap $R_L(N, M, C)$ depends on task similarity $\mathbb{P}-$ and $\mathbb{P}+$ the loss function L. With a loss function that reduces $\text{disc}_L(\mathbb{P}-, \mathbb{P}_+)$, the online-offline discrepancy gap becomes smaller. Our results in Table 2 also show that for some loss functions (e.g., contrastive loss and knowledge distillation loss), the online advantage seems smaller than with other losses like cross-entropy. However, one caveat is that discrepancy distance was originally proposed to deal with binary 0-1 loss and symmetric regression losses (Mansour et al., 2009). With asymmetric losses like cross-entropy, the generalization bound may become looser. Further analysis is needed to account for how common asymmetric losses in deep learning affect the theoretical bounds of continual learning. Nevertheless, current theoretical and empirical results both suggest that the choice of loss function influences the relative performance between online and offline continual learning. Some losses appear to shrink (but not eliminate) the online advantage gap.

Settings. Our empirical results show in task-incremental setting the online advantage gap vanished in task incremental setting. One explanation is the parameter isolation strategy used in task-incremental setting may significantly reduce the discrepancy distance, as suggested by (Ye & Bors, 2022).

Semi-offline CL. We further investigate the generalization ability of semi-offline CL. Corollary 2 shows a higher "patience" λ can lead to higher discrepancy distance. Based on Theorem 1, a higher discrepancy distance increases the generalization bound. This provides some insights on why online CL leads to better performance.

Corollary 2. When $\text{disc}_L(\mathbb{P}_-, \mathbb{P}_+) \neq 0$, $M < N$ and $C > B$, we have:

$$\nabla_\lambda \text{disc}_L(\mathbb{D}, \mathbb{M}) > 0, \forall \lambda \in [0, 1]$$

Proof. $\nabla_\lambda \text{disc}_L = \frac{N^-(N-M)(C-B)}{(N-M_s)^2} \text{disc}_L(\mathbb{P}^-, \mathbb{P}^+)$ □

Based on the proof of Corollary 2, we list a few conditions where the discrepancy distance will not be affected by the "patience" ($\nabla_\lambda \text{disc}_L = 0$), including 1) $N = M$: an unbounded data storage budget to store the whole data stream; 2) $C = B$: the distribution change happens for every batch, i.e., the task size is equal to batch size; 3) $\text{disc}_L(\mathbb{P}_-, \mathbb{P}_+) = 0$: stationary data stream.

6. Conclusion

Many continual learning techniques are applied offline in a task-based manner. Conventional wisdom holds that these methods may fail when applied in an online manner because catastrophic forgetting and underfitting are more difficult to avoid when going through the data in a single pass. This paper challenges that assumption by empirically showing comparable or better performance for online task-free learning given equal memory and compute resources. We corroborate these experimental findings by showing online and offline CL can be unified in a shared algorithmic framework with differences only in the size of long-term and short-term memory and provide theoretical proof that online CL yields a lower generalization bound. We demonstrate that a small short-term memory and a large long-term memory enables more efficient continual learning. By fundamentally rethinking the comparison between online and offline continual learning, we hope this work stimulates further research into unified algorithms optimized for continual knowledge acquisition.

Limitations. This work examines rehearsal and knowledge distillation techniques for continual learning. Other approaches, such as correcting task recency bias (Wu et al., 2019; Hou et al., 2019) or expanding network capacity (Zhou et al., 2022; Yan et al., 2021), are not covered by our analysis. Additionally, we assume random reservoir sampling for exemplar selection. Alternative strategies to construct representative data summaries (Borsos et al., 2020; Bang et al., 2021) may interact differently with the online-offline continuum, presenting another area for empirical analysis through the proposed framework. By broadening the empirical study and adapting the theoretical analysis to incorporate a wider range of continual learning scenarios, the generality of our findings can be strengthened.

7. Impact Statements

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Aljundi, R., Belilovsky, E., Tuytelaars, T., Charlin, L., Caccia, M., Lin, M., and Page-Caccia, L. Online continual learning with maximal interfered retrieval. *Advances in Neural Information Processing Systems*, 32:11849–11860, 2019.

Bang, J., Kim, H., Yoo, Y., Ha, J.-W., and Choi, J. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8218–8227, 2021.

Borsos, Z., Mutny, M., and Krause, A. Coresets via bilevel optimization for continual learning and streaming. *Ad-*

vances in neural information processing systems, 33: 14879–14890, 2020.

Buzzega, P., Boschini, M., Porrello, A., Abati, D., and Calderara, S. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Caccia, L., Aljundi, R., Asadi, N., Tuytelaars, T., Pineau, J., and Belilovsky, E. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*, 2021.

Cai, Z., Sener, O., and Koltun, V. 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, 2021.

Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., and Alahari, K. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 233–248, 2018.

Cha, H., Lee, J., and Shin, J. Co2L: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9516–9525, 2021.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2018.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., and Ranzato, M. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Chrysakis, A. and Moens, M.-F. Online bias correction for task-free continual learning. *ICLR 2023 at OpenReview*, 2023.

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

French, R. M. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Ghunaim, Y., Bibi, A., Alhamoud, K., Alfarra, M., Al Kader Hammoud, H. A., Prabhu, A., Torr, P. H., and Ghanem, B. 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, 2023.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 831–839, 2019.

Huyen, C. *Designing machine learning systems*. " O'Reilly Media, Inc.", 2022.

Jung, D., Lee, D., Hong, S., Jang, H., Bae, H., and Yoon, S. New insights for the stability-plasticity dilemma in online continual learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Li, H., Jiang, H., Gu, X., Peng, J., Li, W., Hong, L., and Tao, C. CLRS: Continual learning benchmark for remote sensing image scene classification. *Sensors*, 20(4):1226, 2020.

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

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Ma, X., Jeong, S., Zhang, M., Wang, D., Choi, J., and Jeon, M. Cost-effective on-device continual learning over memory hierarchy with miro. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pp. 1–15, 2023.

Mai, Z., Li, R., Kim, H., and Sanner, S. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3589–3599, 2021.

Mai, Z., Li, R., Jeong, J., Quispe, D., Kim, H., and Sanner, S. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

Mallya, A. and Lazebnik, S. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation: Learning bounds and algorithms. In *The 22nd Conference on Learning Theory*, 2009. URL http://www.cs.mcgill.ca/\%7Ecolt2009/papers/003.pdf\#page=1.

Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D., and Van De Weijer, J. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022.

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

Montiel, J., Read, J., Bifet, A., and Abdessalem, T. Scikit-multiflow: A multi-output streaming framework. *Journal of Machine Learning Research*, 19(72):1–5, 2018.

Prabhu, A., Al Kader Hammoud, H. A., Dokania, P. K., Torr, P. H., Lim, S.-N., Ghanem, B., and Bibi, A. Computationally budgeted continual learning: What does matter? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3698–3707, 2023.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Serra, J., Suris, D., Miron, M., and Karatzoglou, A. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pp. 4548–4557. PMLR, 2018.

Soutif-Cormerais, A., Carta, A., Cossu, A., Hurtado, J., Lomonaco, V., Van de Weijer, J., and Hemati, H. A comprehensive empirical evaluation on online continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3518–3528, 2023.

Van de Ven, G. M. and Tolias, A. S. Three scenarios for continual learning. *NeurIPS - Continual Learning workshop*, 2019.

Vitter, J. S. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1): 37–57, 1985.

Wang, L., Zhang, X., Yang, K., Yu, L., Li, C., Lanqing, H., Zhang, S., Li, Z., Zhong, Y., and Zhu, J. Memory replay with data compression for continual learning. In *International Conference on Learning Representations*, 2021.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 374–382, 2019.

Yan, S., Xie, J., and He, X. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3014–3023, 2021.

Ye, F. and Bors, A. G. Task-free continual learning via online discrepancy distance learning. *Advances in Neural Information Processing Systems*, 35:23675–23688, 2022.

Zhang, Y., Pfahringer, B., Frank, E., Bifet, A., Lim, N. J. S., and Jia, Y. A simple but strong baseline for online continual learning: Repeated augmented rehearsal. *Advances in Neural Information Processing Systems*, 35: 14771–14783, 2022.

Zhou, D.-W., Wang, Q.-W., Ye, H.-J., and Zhan, D.-C. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *The Eleventh International Conference on Learning Representations*, 2022.

A. Proofs

A.1. Proof of Proposition 1

Proof. Let $\gamma \doteq \frac{N^-}{N}$ and $\alpha \doteq \frac{M_s}{M}$. Based on the definition of discrepancy distance, we have:

$$\begin{aligned}&\mathrm{disc}_L(\mathbb{D},\mathbb{M})\\ &= \max_{h,h'\in H} \big|\gamma\mathcal{L}_{\mathbb{P}^-}(h',h) + (1-\gamma)\mathcal{L}_{\mathbb{P}^+}(h',h)\\ &\qquad - \big((1-\alpha)\mathcal{L}_{\mathbb{M}_{long}}(h',h) + \alpha\mathcal{L}_{\mathbb{M}_{short}}(h',h)\big)\big|\end{aligned} \tag{9}$$

Since the long-term memory is managed by the reservoir sampling method, and letting $\beta \doteq \frac{N^-}{N-\alpha M}$, we have $\mathcal{L}_{\mathbb{M}_{long}}(h',h) = \beta\mathcal{L}_{\mathbb{P}^-}(h',h) + (1-\beta)\mathcal{L}_{\mathbb{P}^+}(h',h)$ and $\mathcal{L}_{\mathbb{M}_{short}}(h',h) = \mathcal{L}_{\mathbb{P}^+}(h',h)$. Inserting these two results into Eq 9 gives:

$$\begin{aligned}&\mathrm{disc}_L(\mathbb{D},\mathbb{M})\\ &= \max_{h,h'\in H} \big|(\gamma-(1-\alpha)\beta)\big)\left(\mathcal{L}_{\mathbb{P}^+}(h',h) - \mathcal{L}_{\mathbb{P}^-}(h',h)\right)\big|\\ &= (\gamma-(1-\alpha)\beta)\big) \max_{h,h'\in H} \big|\left(\mathcal{L}_{\mathbb{P}^+}(h',h) - \mathcal{L}_{\mathbb{P}^-}(h',h)\right)\big|\end{aligned} \tag{10}$$

This last equality is based on the fact that $\gamma-(1-\alpha)\beta = \frac{\alpha N^-(N-M)}{N(N-\alpha M)} > 0$ when $N > M$. □

A.2. Proof of Corollary 1

Proof. Since in offline CL, the short-term memory size is task size: $M_s = C$, inserting this and $N^- = N - C$

(definitions) in Eq 7, we have

$$\begin{aligned}\text{disc}_L(\mathbb{D}, \mathbb{M}_{offline}) &= \frac{N^- C(N-M)}{NM(N-C)} \text{disc}_L(\mathbb{P}_-, \mathbb{P}_+) \\ &= \frac{C(N-M)}{NM} \text{disc}_L(\mathbb{P}_-, \mathbb{P}_+)\end{aligned} \tag{11}$$

In online CL, the short-term memory size is the incoming batch size: $M_s = B$.Inserting this in Eq 7, we have

$$\text{disc}_L(\mathbb{D}, \mathbb{M}_{online}) = \frac{N^- B(N-M)}{NM(N-B)} \text{disc}_L(\mathbb{P}_-, \mathbb{P}_+) \tag{12}$$

Combining Eq 11 and Eq 12 two gives us corollary 1. □

B. Pseudo-code of online and offline CL

Algorithm 2: Online vs. Offline ER

```
function Offline ER
    for each task data 𝒞 do
        // Model update with K epochs
        for epoch = 1,...,K do
            for 𝒳 in 𝒞 do
                𝒳_long ~ M_long,
                θ ← θ − η∇L(𝒳 ∪ 𝒳_long; θ)
        // Exemplar update
        M_long ←_π 𝒞
function Online ER
    for each batch data 𝒳 do
        // Model update with K
           gradients
        for iter = 1,...,K do
            𝒳_long ~ M_long,
            θ ← θ − η∇L(𝒳 ∪ 𝒳_long; θ)
        // Exemplar update
        M_long ←_π 𝒳
```

C. Dataset and Experiment details

Table 5 lists the image size, the number of classes, the number of tasks, and data size per task for the four CL benchmarks.

Training details Following (Masana et al., 2022), all experiments utilized ResNet-18 with a single head. We use standard data augmentation (random cropping and flipping), a data stream batch size of 50 for CIFAR100 and Mini-Imagenet, 64 for CORE50. A equal number of exemplars are sampled at each gradient step. All models use vanilla SGD for optimization with a learning rate of 0.1. For iCaRL and SCR, a nearest-class-mean (NCM) classifier is applied as in the original publications. The default iteration and epoch number is 50. We run all experiments across three random seeds. Other hyperparameter details are presented in Appendix C.

Hyperparameters. The hyperparameter settings are summarized in Table 6. The regularization strength in DER++ and temperature values in SCR follow the original papers.

D. Additional experiment results

The influence of compute on Stability and Plasticity for different CL algorithms is shown in Fig 7 and 6. In most cases, increasing compute leads to a higher plasticity with lower stability. However, in a few cases, online ER (iteration 1-10) and online and offline DER (iteration 1-10), increased compute leads to higher plasticity and stability at the same time. In some other cases, offline DER (50-100), higher compute leads to lower plasticity but higher stability.

The stability and plasticity dynamics for Split CIFAR100 is shown in Fig 8.

Table 4. The compute and data storage cost of online and offline continual learning on the same CL benchmark Split-CIFAR100.

	Papers	Data Storage (# samples)		Compute
		Exemplar	Working buffer	Iteration/Epoch
online CL	ER (Chaudhry et al., 2019)	2000-5000	10	1
	ER-ACE (Caccia et al., 2021)			1
	SCR (Mai et al., 2021)			1
	ER-OBC (Chrysakis & Moens, 2023)			1
	RAR (Zhang et al., 2022)			10
	Survey (Mai et al., 2022)			1,5
	Survey (Soutif-Cormerais et al., 2023)			3
offline CL	ICARL (Rebuffi et al., 2017)	2000-5000	2500-5000	70
	EEIL (Castro et al., 2018)			70
	LUCIR (Hou et al., 2019)			160
	DER++ (Buzzega et al., 2020)			50
	MEMO (Zhou et al., 2022)			170
	BIC (Wu et al., 2019)			250
	Survey (Masana et al., 2022)			100

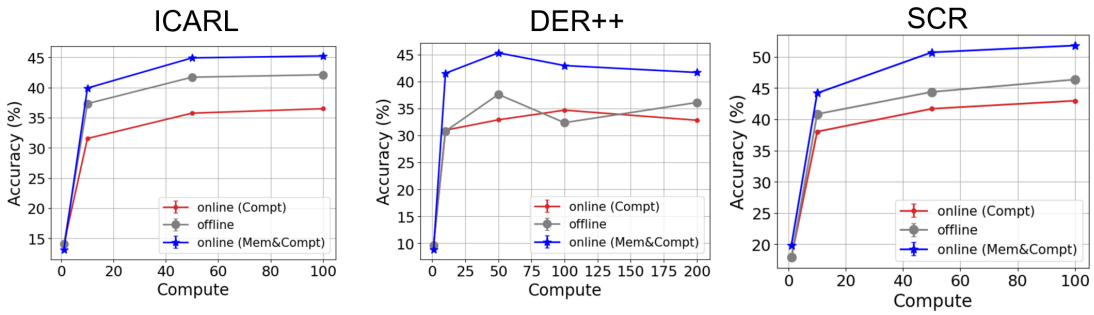


Figure 6. The *online replay advantage* in different CL algorithms for Split Mini-ImageNet

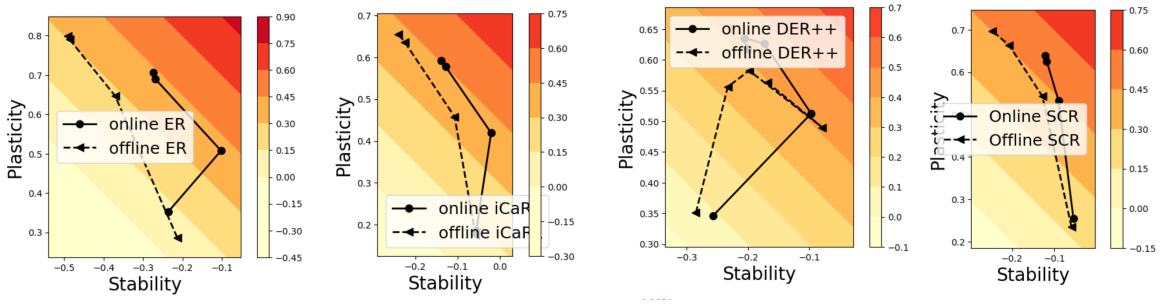


Figure 7. Insights on how *online replay advantage* occurs. In some cases (ER, iCaRL, SCR), online CL leads to greater stability than offline CL; In other case (DER++), online CL improves plasticity over offline CL.

Table 5. Dataset information for the four CL benchmarks.

	IMAGE SIZE	#TASK	# CLASS	TRAIN PER TASK	TEST PER TASK
SPLIT-CIFAR100	3X32X32	10	100	5,000	500
SPLIT-MINI-IMAGENET	3X84X84	10	100	5,000	1,000
SPLIT-CORE50-NC	3X128X128	9	50	12,000	4,500
CLRS-NI (DOMAIN-INCREMENTAL)	3X256X256	5	25	5375	3750
SPLIT-IMAGENET-1K	3X224X224	10	1000	∼120,000	5000

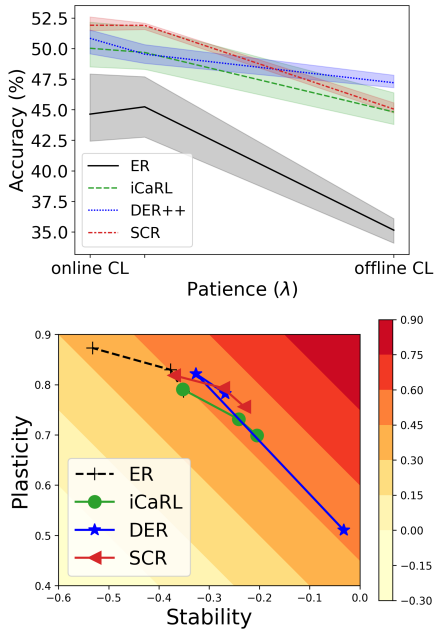


Figure 8. Semi-offline results for Split-CIFAR100.

Table 6. Hyperparameter setting.

	HYPERPARAMETER
ER	LR=0.1
ICARL	LR=0.1,NCM CLASSIFIER
SCR	TEMP =0.07, LR=0.1, NCM CLASSIFIER
DER ++	$\alpha = 0.1$, $\beta = 0.5$,$lr = 0.03$ (CIFAR100)
	$\alpha = 0.3$ $\beta = 0.8$,$lr = 0.1$ (MINI-IMAGENET)
	$\alpha = 0.1$, $\beta = 1.0$,$lr = 0.1$ (CORE50)

Table 7. Accuracy of aligned online and offline continual learning methods in ImageNet1000 for increments of 100 classes. Offline ER uses 20,000 exemplars, and online ER uses 120,000 exemplars. Due to the compute time constraint, we report the results of the first 500 classes. Full results are shown in Figure 9.

# CLASSES	100	200	300	400	500
ER OFFLINE	81.6	50.4	48.0	43.1	39.4
ER ONLINE	82.5	70.2	62.0	60.0	58.5

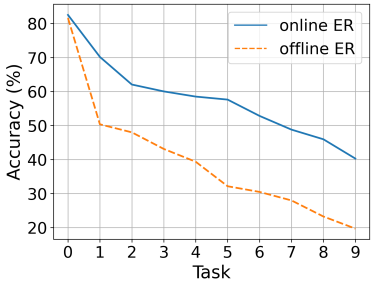


Figure 9. Comparing online and offline ER for ImageNet with 50 iteration budget in setting 2 (aligned compute and storage). Online ER outperforms offline ER in this large-scale datasets.

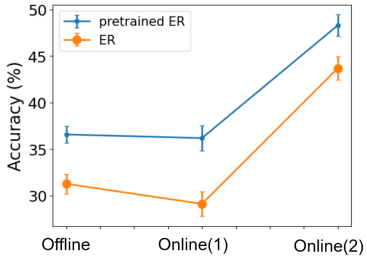


Figure 10. Comparing online and offline ER using pretrained-ResNet18 in Mini-ImageNet datasets.

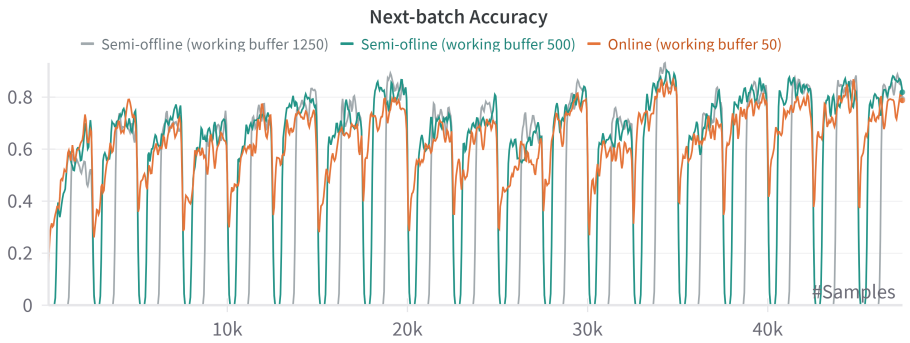


Figure 11. Next-batch evaluation of semi-offline cases.Gaussian smoothing is employed for better readability. The experiment is conducted on CIFAR100 with 50 iteration/epochs.