Continual Learning: Applications and the Road Forward

Anonymous authors
Paper under double-blind review

Abstract

Continual learning is a sub-field of machine learning, which aims to allow machine learning models to continuously learn on new data, by accumulating knowledge without forgetting what was learned in the past. In this work, we take a step back, and ask: "*Why should one care about continual learning in the first place?*". We set the stage by surveying recent continual learning papers published at three major machine learning conferences, and show that memory-constrained settings dominate the field. Then, we discuss five open problems in machine learning, and even though they seem unrelated to continual learning at first sight, we show that continual learning will inevitably be part of their solution. These problems are model-editing, personalization, on-device learning, faster (re-)training and reinforcement learning. Finally, by comparing the desiderata from these unsolved problems and the current assumptions in continual learning, we highlight and discuss four future directions for continual learning research. We hope that this work offers an interesting perspective on the future of continual learning, while displaying its potential value and the paths we have to pursue in order to make it successful.

1 Introduction

Continual learning, sometimes referred to as lifelong learning or incremental learning, is a sub-field of machine learning that focuses on the challenging problem of incrementally training models on a stream of data with the aim of accumulating knowledge over time. This setting calls for algorithms that can learn new skills with minimal forgetting of what they had learned previously, transfer knowledge across tasks, and smoothly adapt to new circumstances when needed. This is in contrast with the traditional setting of machine learning, which typically builds on the premise that all data, both for training and testing, are sampled i.i.d. (independent and identically distributed) from a single, stationary data distribution.

Deep learning models in particular are in need of continual learning capabilities. A first reason for this is their strong dependence on data. When trained on a stream of data whose underlying distribution changes over time, deep learning models tend to adapt to the most recent data, thereby "catastrophically" forgetting the information that had been learned earlier (French, 1999). Secondly, continual learning capabilities could reduce the very long training times of deep learning models. When new data are available, current industry practice is to retrain a model fully from scratch on all, past and new, data (see Example 3.4). Such retraining is time inefficient, sub-optimal and unsustainable, with recent large models exceeding 10.000 GPU days of training (Radford et al., 2021). Simple solutions, like freezing feature extractor layers, are often not an option as the power of deep learning hinges on the representations learned by those layers (Bengio et al., 2013). To work well in challenging applications in e.g. computer vision and natural language processing, they often need to be changed.

The paragraph above describes two naive approaches to the continual learning problem. The first one, incrementally training – or finetuning – a model only on the new data, usually suffers from suboptimal performance when models adapt too strongly to the new data. The second approach, repeatedly retraining a model on all data used so far, is undesirable due to its high computational and memory costs. The goal of continual learning is to find approaches that have a better trade-off between performance and efficiency (e.g. compute and memory) than these two naive ones. In the contemporary continual learning literature, this trade-off typically manifests itself by limiting memory capacity and optimizing performance under this constraint. Computational costs are not often considered in the current continual learning literature, although this is challenged in some recent works, which we discuss in Sections 2 and 4.1.

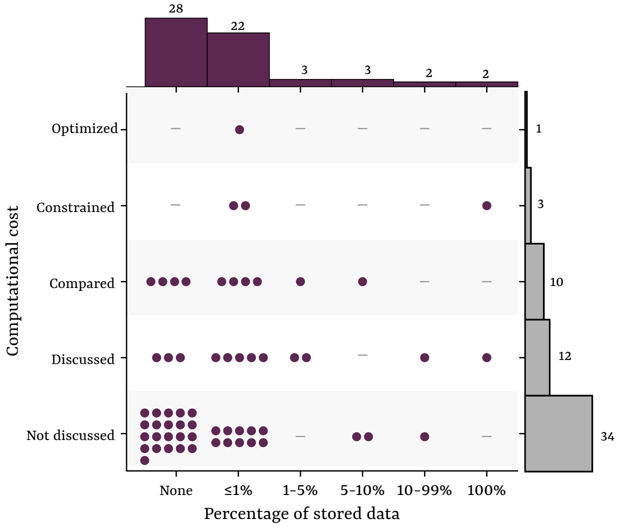


Figure 1: **Most papers strongly restrict memory use and do not discuss computational cost**. The figure shows an overview of the surveyed papers in Section 2. Each dot represents one paper, illustrating what percentage of data their methods store (horizontal axis) and how computational cost is handled (vertical axis). The majority of surveyed papers are in the lower-left corner: those that strongly restrict memory use and do not quantitatively approach computational cost (i.e. it is at most discussed). For more details, see Appendix.

In this article, we highlight several practical problems in which there is an inevitable continual learning component, often because there is some form of new data that is available for a model to train on. We discuss how these problems require continual learning, and how in these problems that what is constrained and that what is optimized differs. Constraints are hard limits set by the environment of the problem (e.g. small devices have limited memory), under which other aspects, such as computational cost and performance, need to be optimized. Progress in the problems we discuss goes hand in hand with progress in continual learning, and we hope that they serve as a motivation to continue working on continual learning, and offer an alternative way to look at it and its benefits. Similarly, they can offer an opportunity to align currently common assumptions that stem from the benchmarks we use, with those derived from the problems we aim to solve. Section 3 describes some of these problems and in Section 4 we discuss some exciting future research directions in continual learning, by comparing the desiderata of the discussed problems and contemporary continual learning methods.

2 Current continual learning

Before exploring different problem settings in which we foresee continual learning as a useful tool, we first wish to understand the current landscape. Our aim is to paint a clear picture of how memory and computational cost are approached. To achieve this, we surveyed continual learning papers accepted at three top machine learning conferences (ECCV '22, NeurIPS '22 and CVPR '23). We considered all papers with either *'incremental', 'continual', 'forgetting', 'lifelong'* or *'catastrophic'* in their titles, disregarding false positives. See Appendix for the methodology. For our final set of 60 papers, we investigated how they balance the memory and compute cost trade-offs. We discern five categories:

Not discussed: No clear mention of the impact of the proposed method/analysis on the cost

Discussed: Cost is discussed in text, but not quantitatively compared between methods.

Compared: Cost is qualitatively compared to other methods

Constrained: Methods are compared using the same limited cost.

Optimized: Cost is among the optimized metrics.

Many continual learning papers use memory in a variety of ways, most often in the form of storing samples, but regularly model copies (e.g. for distillation) or class means and their variances are stored as well. We focus on the amount of stored data, as this is the most common use of memory, but discuss other memory costs in the Appendix. Of the surveyed papers, all but two constrain the amount of stored samples. So rather than reporting the category, in Figure 1, we report how strongly it is constrained, using the percentage of all data that is stored. It is apparent that the majority of these papers do not store any (raw) samples and many are using only a small fraction. Two notable exceptions that store all the raw data are a paper on continual reinforcement learning (RL) (Fu et al., 2022), something which is not uncommon in RL, see Section 3.5. The second one, by Prabhu et al. (2023a), studies common CL algorithms under a restricted computational cost.

While memory costs (for raw samples) are almost always constrained, computational costs are much less so. Sometimes simply discussing that there is (almost) no additional computational cost can suffice, yet it is remarkable that in more than 50% of the papers there is no mention of the computational cost at all. When it is compared, it is often done in the appendix. There are a few notable exceptions in the survey, which focus explicitly on the influence of the computational cost, either by constraining (Prabhu et al., 2023a; Kumari et al., 2022; Ghunaim et al., 2023) or optimizing it (Wang et al., 2022b). For a more elaborate discussion of measuring the computational cost, see Section 4.1. Together, these results show that many continual learning methods are developed with a low memory constraint, and with limited attention to the computational cost. They are two among other relevant dimensions of continual learning in biological systems (Kudithipudi et al., 2022) and artificial variants (Mundt et al., 2022), yet with the naive solutions of the introduction in mind, they are two crucial components of any continual learning algorithm. In the next section, we introduce some problems for which continual learning is inevitable. They illustrate that methods with a low computational cost is just as well an important setting, yet it has not received the same level of attention.

3 Continual learning is not a choice

To solve the problems described in this section, continual learning is necessary and not just a tool that one could use. We argue that in all of them, the problem can, at least partly, be recast as a continual learning problem. This means that the need for continual learning algorithms arises from the nature of the problem itself, and not just from the choice of a specific way for solving it. We start these subsections by explaining what the problem is and why it fundamentally requires continual learning. Next we briefly discuss current solutions and how they relate to established continual learning algorithms. We conclude each part by laying down what the constraints are and what metrics should be optimized.

3.1 Adapting machine learning models locally

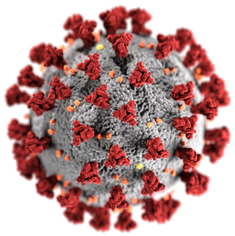
It is often necessary to correct wrongly learned predictions from past data. Real world practice shows us that models are often imperfect, e.g. models frequently learn various forms of decision shortcuts (Lapuschkin et al., 2019), or sometimes the original training data become outdated and are no longer aligned with current facts (e.g. a change in government leaders). Additionally, strictly accumulating knowledge may not always be compliant with present legal regulations and social desiderata. Overcoming existing biases, more accurately reflecting fairness criteria, or adhering to privacy protection regulations (e.g. the right to be forgotten of the GDPR in Europe (Union, 2016)), represent a second facet of the editing problem.

When mistakes are exposed, it is desirable to selectively edit the model without forgetting other relevant knowledge and without re-training from scratch. The model editing pipeline (Mitchell et al., 2022) first identifies corner cases and failures, then prompts data collection over those cases, and subsequently re-trains/updates the model. Recently proposed methods are able to locally change models, yet this comes at a significant cost, or model draw-down, i.e. forgetting of knowledge that was correct (Santurkar et al., 2021). Often the goal of model editing is to change the output associated with a specific input from A to B, yet changing the output to something generic or undefined is an equally interesting case. Such changes can be important in privacy-sensitive applications, to e.g. forget learned faces or other personal attributes.

Naively, one could retrain a model from scratch with an updated dataset, that no longer contains outdated facts and references to privacy-sensitive subjects, or includes more data on previously out-of-distribution data. To fully retrain on the new dataset, significant computational power and access to all previous training data is necessary. Instead, with effective continual learning, this naive approach can be improved by only changing what should be changed. An ideal solution would be able to continually fix mistakes, at a much lower computational cost than retraining from scratch, without forgetting previously learned and unaffected information. Such a solution would minimize computational cost, while maximizing performance. There is no inherent limitation on memory in this problem, although it can be limited if not all training data are freely accessible.

Example: 3.1

Lazaridou et al. (2021) used the customnews benchmark to evaluate how well a language model trained on news data from 1969 – 2017 performs on data from 2018 and 2019. They find that models perform worse on the newest data, mostly on proper nouns (e.g. "Ardern" or "Khashoggi"), as well as words introduced because of societal changes such as "Covid-19" and "MeToo". They identify a set of 287 new words that were not used in any document prior to 2018. Such new words are inevitable in future texts too. To teach a model these changes they perform updates on the newly arriving data, which gradually improves the performance on the years 2018 and 2019 (a 10% decrease in perplexity), yet at the cost of performance on earlier years (a 5% increase on *all* previous years). When weighing all years equally, the final model thus got worse than before updating.



3.2 Incorporation of user- and domain-specific knowledge

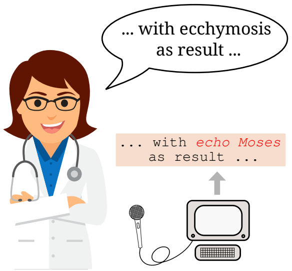
Some of the most powerful machine learning models are trained on very large datasets, usually scraped from the Internet. The result is a model that is able to extract useful and diverse features from high-dimensional data. However, the vastness of the data they are trained on also has a downside. Internet data is generated by many different people, who all have their own preferences and interests. One model cannot fit these conflicting preferences, and the best fit is close to the average internet user (Hu et al., 2022b). However, machine learning models are often used by individuals or small groups, or for highly specific applications. This contradiction makes any possessive references such as 'my car' or 'my favorite band' by construction ambiguous and impossible for the system to understand. Further, Internet scraped data often do not contain (enough) information to reach the best performance in specialized application domains like science and user sentiment analysis (Beltagy et al., 2019).

Domain adaptation and personalization are thus often necessary. The topic has been investigated in the natural language processing (NLP) community for many different applications. Initially, fine-tuning on a supervised domain-specific dataset was the method of choice, but recently, with the success of very large language models (LLM), the focus has shifted towards changing only a small subset of parameters with adapters (Houlsby et al., 2019), low-rank updates (Hu et al., 2022a) or prompting (Jung et al., 2023). However, these methods do not explicitly identify and preserve important knowledge in the original language model. This hampers the integration of general and domain-specific knowledge and produces weaker results (Ke et al., 2022). To identify the parameters that are important for the general knowledge in the LLM in order to protect them is a challenging problem. Recent works (Ke et al., 2021) made some progress in balancing the trade-off between performance on in-domain and older data. In the computer vision field, similar work has also been done by adapting CLIP to different domains (Wortsman et al., 2022) and to include personal text and image pairs (Cohen et al., 2022).

No matter how large or sophisticated the pre-trained models become, there will always be data that they are not, or cannot be, trained on (e.g. tomorrow's data). It is impossible to acquire all the information in the world, and even if it were possible, that cannot result in personalized models. When specialized data are collected afterwards, models can be updated either on the original machine or on a smaller device. Again, the final goal is to train a specialized or personalized model, more compute-efficient than when trained from scratch. On the original training server, past data are usually available. When this is not the case, because training happens on a more restricted (e.g. personal) device, memory does become a constraint, which we elaborate on in the next subsection.

Example: 3.2

Dingliwal et al. (2023) personalize end-to-end speech recognition models with words that are personal to the user (e.g. family member names) or words that are very rare except in specialized environments (e.g. "ecchymoses" in medical settings). With an extra attention module and a pre-computed set of representations of the specialized vocabulary, they 'bias' the original model towards using the new rare and unknown words. The performance on the specialized words is remarkably improved, yet with a decrease in performance on non-specialist word recognition. In their experiments specialized tokens are less than 1% off all tokens, so even a relatively small decrease in performance on other tokens is non-negligible.


... with ecchymosis as result ...
... with echo Moses as result ...

3.3 On-device learning

To offer an experience aligned with a user's preferences, or adjusted to a new personal environment, many deep learning applications require updates on the deployed device. Cloud computing is often not available because of communication issues (e.g. in remote locations with restricted internet access, or when dealing with very large quantities of data), or to preserve the privacy of the user (e.g. for domestic robots, monitoring cameras). On such small devices, both memory and computational resources are typically constrained, and the primary goal is to maximize model efficacy under these constraints. These tight constraints often make storing all user data and retraining from scratch infeasible, necessitating continual learning whenever the pre-trained capabilities should not be lost during continued on-device training (see also Example 3.3).

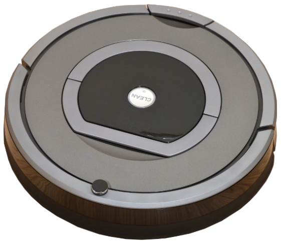
These constraints, as well as increasingly complex computations for energy-accuracy trade-offs in real-time (Kudithipudi et al., 2023), limit the direct application of optimization typically used in cloud deployments. For example, existing methods only update the final classification layer of a pre-trained feature extractor (Hayes & Kanan, 2022). Yet this relatively lightweight process becomes challenging when there is a large domain gap between the initial training set and the on-device data. The latter is often hard to collect, since labeling large amounts of data by the user is impractical, requiring few-shot solutions. When devices shrink even further, the communication costs become significant, and reading and writing to memory can be up to ∼99% of the total energy budget (Dally, 2022). In addition to algorithmic optimizations for continual learning, architectural optimizations offer interesting possibilities. These enhancements may include energy-efficient memory hierarchies, adaptable dataflow distribution, domain-specific compute optimizations like quantization and pruning, and hardware-software co-design techniques (Kudithipudi et al., 2022).

On-device learning from data that is collected locally almost certainly involves a distribution shift from the original (pre-)training data. This means the sampling process is no longer i.i.d., thus requiring continual learning to maintain good performance on the initial training set. If these devices operate on longer time scales, the data they sample themselves will not be i.i.d. either. To leverage the originally learned information as well as adapt to local distribution changes, such devices require continual learning to operate effectively. Importantly, they should be able to learn using only a limited amount of labeled information, while operating under the memory and compute constraints of the device.

3.4 Faster retraining with warm starting

In many industrial settings, deep neural networks are periodically re-trained from scratch when new data are available, or when a distribution shift is detected. The newly gathered data is typically a lot smaller than the original dataset is, which makes starting from scratch a wasteful endeavor. As more and more data is collected, the computational requirements for retraining continue to grow over time. Instead, continual learning can start from the initial model and only update what is necessary to improve performance on the

Example: 3.3



In a 2022 article by MIT Review (Guo, 2022), it was revealed how a robot vacuum cleaner had sent images, in some cases sensitive ones, back to the company, to be labeled and used in further training on central servers. In response, an R&D director of the company stated: *"Road systems are quite standard, so for makers of self-driving cars, you'll know how the lane looks [...], but each home interior is vastly different"*, acknowledging the need to adjust the robots to the environment they are working in. Our homes are highly diverse, but also one of the most intimate and private places that exist. Images can reveal every detail about them and should thus remain private. Adapting to individual homes is necessary, but should not come at the cost of initial smart abilities such as object recognition, collision prevention and planning, which are unlikely to be learned using only locally gathered data.

new dataset. Most continual learning methods are not designed for computational efficiency (Harun et al., 2023a), yet Harun et al. (2023b) show that reductions in training time by an order of magnitude are possible, while reaching similar performance. Successful continual learning would offer a way to drastically reduce the expenses and extraordinary carbon footprint associated with retraining from scratch (Amodei & Hernandez, 2018), without sacrificing accuracy.

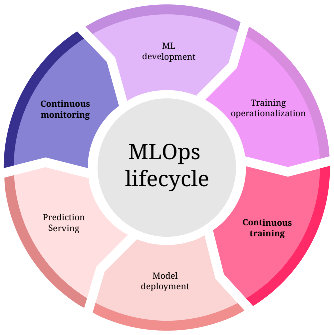
The challenge is to achieve performance equal to or better than a solution that is trained from scratch, but with fewer additional resources. One could say that it is the performance that is constrained, and computational cost that must be optimized. Simple approaches, like warm-starting, i.e. from a previously trained network, can yield poorer generalization than models trained from scratch on small datasets (Ash & Adams, 2020), yet it is unclear whether this translates to larger datasets, and remains a debated question. Similar results were found in (Berariu et al., 2021; Dohare et al., 2023), which report a loss of plasticity, i.e. the ability to learn new knowledge after an initial training phase. In curriculum learning (Bengio et al., 2009), recent works have tried to make learning more efficient by cleverly selecting which samples to train on when (Hacohen et al., 2020). Similarly, active learning (Settles, 2009) studies which unlabeled samples could best be labeled (given a restricted budget) to most effectively learn. Today those fields have to balance learning new information with preventing forgetting, yet with successful continual learning they could focus more on learning new information as well and as quickly as possible.

Minimizing computational cost could also be rephrased as maximizing learning efficiency. Not having to re-learn from scratch whenever new data is available, figuring out the best order to use data for learning, or the best samples to label can all contribute to this goal. Crucially, maximizing knowledge accumulation from the available data is part of this challenge. Previous work (Hadsell et al., 2020; Hacohen et al., 2020; Pliushch et al., 2022) suggested that even when all data is used together, features are learned in sequential order. Exploiting this order to make learning efficiently requires continual learning.

3.5 Reinforcement learning

In reinforcement learning (RL), agents learn by interacting with an environment. This creates a loop, where the agent takes an action within the environment, and receives from the environment an observation and a reward. The goal of the learning process is to learn a policy, i.e. a strategy to choose the next action based on the observations and rewards seen so far, which maximizes the rewards (Sutton & Barto, 2018). Given that observations and rewards are conditioned on the policy, this leads to a natural non-stationarity, where each improvement step done on the policy can lead the agent to explore new parts of the environment. The implicit non-stationarity of RL can be relaxed to a piece-wise stationary setting in off policy RL settings Sutton & Barto (2018), however this still implies a continual learning problem. Offline RL (Levine et al., 2020) (e.g. imitation learning) completely decouples the policy used to collect data from the learning policy, leading to a static data distribution, though is not always applicable and can lead to suboptimal solutions due to the inability of the agent to explore. Lastly, for real-world problems, the environment itself may be non-stationary, either intrinsically so, or through the actions of the agent.

Example: 3.4



Continuous training is one of the six important building blocks in MLOps (Machine Learning Operations, similar to DevOps), according to a Google white paper on the subject (Salama et al., 2021). This step is considered necessary, in response to performance decays when incoming data characteristics change. They describe in great detail how to optimize this pipeline, from various ways to trigger retraining to automated approaches to deploy retrained models. However, retraining is implicitly considered to be from scratch, which makes most pipelines inherently inefficient. Similarly, other resources stating the importance of retraining ML models and efficient MLOps, at most very briefly consider other options than retraining from scratch (Kreuzberger et al., 2023; Komolafe, 2023; Alla et al., 2021). The efficiency that can be gained here represents an enormous opportunity for the continual learning field, which is clearly illustrated by Huyen (2022) from an industry perspective.

The presence of non-stationarities in reinforcement learning makes efficient learning difficult. To accelerate learning, experience replay has been an essential part of reinforcement learning (Lin, 1992; Mnih et al., 2015). While engaging in new observations, previously encountered states and action pairs are replayed to make training more i.i.d. In contrast to replay in supervised learning, in RL there is less focus on restricting the amount of stored examples, as the cost of obtaining them is considered very high. Instead the focus is how to select samples for replay (e.g. Schaul et al., 2016) and how to create new experiences from stored ones (Lin et al., 2021). Additionally, loss of plasticity (e.g. Dohare et al., 2023; Lyle et al., 2022) — inability of learning efficiently new tasks — and formalizing the concept of continual learning (e.g. Kumar et al., 2023; Abel et al., 2023) also take a much more central role in the RL community.

Finally, besides the non-stationarities encountered while learning a single task, agents are often required to learn multiple tasks. This setting is an active area of research (Wołczyk et al., 2021; Kirkpatrick et al., 2017; Rolnick et al., 2019), particularly since the external imposed non-stationarity allows the experimenter to control it and probe different aspects of the learning process. RL has its own specific problems with continual learning, e.g. trivially applying rehearsal methods fails in the multi-task setting, and not all parts of the network should be regularized equally (Wolczyk et al., 2022). Issues considering the inability to learn continually versus the inability to explore an environment efficiently, as well as dealing with concepts like episodic and non-episodic RL, makes the study of continual learning in RL more challenging. Further research promises agents that train faster, learn multiple different tasks sequentially and effectively re-use knowledge from previous tasks to work faster and towards more complex goals.

Example: 3.5

Typical RL methods store millions or more transitions in a replay memory. Schaul et al. (2016) showed that theoretically exponential training speed-ups are possible when cleverly selecting the transitions to replay. By approximating 'how much the model can learn' from a transition, they prioritize some samples over others and practically show a linear speed-up compared to uniform selection, the default at that point. Current state-of-the-art in the Atari-57 benchmark, MuZero (Schrittwieser et al., 2020), relies on this prioritized selection and confirms its importance, yet from the initial theoretical results, it is clear that improved continual learning could further improve convergence speeds and results (e.g. Pritzel et al., 2017).

4 Future directions for continual learning

In this section we discuss interesting future directions for continual learning research, informed by what was discussed in the previous sections. We start by addressing the motivation for these research directions, followed by a brief overview of existing work, and finally justifying the importance of each concept.

4.1 Rethinking memory and compute assumptions

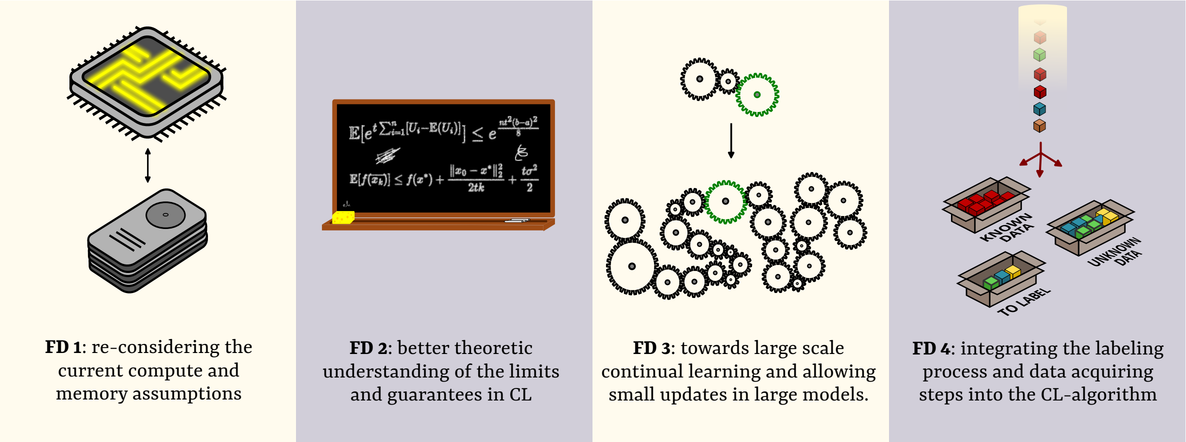
In all of the problems described in the previous section, optimizing or restricting compute complexity plays an important role, often a more central one than memory capacity does. This is in stark contrast to the results of the survey in Section 2. The vast majority of papers does not qualitatively approach compute complexity, while not storing, or only very few, samples. Two popular reasons for arguing a low storage solution are the cost of memory and privacy concerns, but these arguments are often not relevant in practice. Prabhu et al. (2023b) calculate that the price to store ImageNet1K for one month is just 66¢, while training a model on it requires 500$. This means storing the entire dataset for 63 years is as expensive as training ImageNet once. Further, privacy and copyright concerns are not solved by simply deleting data from the training set, as data can be recovered from derivative models (Haim et al., 2022), and rulings to remove data might only be viable by re-training from scratch (Zhao, 2022) (hence making continual learning superfluous), at least until reliable model editing exists (see Section 3.1). Section 3.3 showed that use cases in low memory settings exist, but, as the four of the five problems show, there are many reasons to study algorithms that restrict computational cost just like restricted memory settings are studied today. We believe it is important to reconsider these common assumptions on memory and computational cost, and instead derive them from the real-world problems that continual algorithms aim to solve.

To achieve this goal, we should agree on how to measure computational cost, which is necessary to restrict it. Yet it is not straightforward to do so. Recent approaches use the number of iterations (Prabhu et al., 2023a) and forward/backward passes (Kumari et al., 2022; Harun et al., 2023c), which works well if the used model is exactly the same, but cannot capture architectural differences that influence computational cost. Similarly, when the number of parameters is used (Wang et al., 2022a), more iterations or forward passes do not change the perceived cost. The number of floating point operations (FLOPs) is often used to measure computational cost in computer science, and is a promising candidate, yet is sometimes hard to measure accurately (Wang et al., 2022b). Additionally, time to convergence should also be considered, as faster convergence would also lower compute time. See Schwartz et al. (2020) for an elaborate discussion. To properly benchmark compute time and memory use in continual learning algorithms, we should build on this existing literature to attain strong standards for measuring both compute and memory cost and the improvements thereof.

As illustrated in Section 2, there are works that have started to question our common assumptions in continual learning (Harun et al., 2023b). SparCL optimizes compute time explicitly (Wang et al., 2022b), while (Prabhu et al., 2023b;a) compare methods while constraining computational cost. Chavan et al. (2023) establish DER (Yan et al., 2021) as a promising method when compute complexity is constrained, while other works suggest that experience replay is likely most efficient (Harun et al., 2023c; Prabhu et al., 2023a). These early works have laid the groundwork, and we believe that it is in continual learning's best interest to push further in this direction, and develop strategies for learning under a tight compute budget, with and especially *without* memory constraints.

4.2 Theory

In the past, continual learning research has achieved interesting empirical results. In contrast to classic machine learning, not much is known about whether and under which conditions, we can expect results. Many theoretical results rely on the i.i.d assumption, among which the convergence of stochastic gradient descent and the difference between expected and empirical risk in many PAC-bound analyses (although there are some exceptions, e.g. Pentina & Lampert 2014). Crucially, the i.i.d. assumption is almost always broken in continual learning, as illustrated by the problems in Section 3. To a certain extent this also happens in training with very large datasets, due to the computational cost of sampling data batches in an i.i.d.


$\mathbb{E}[e^{t\sum_{i=1}^{n}[U_i - \mathbb{E}(U_i)]}] \leq e^{\frac{nt^2(b-a)^2}{8}}$
$\mathbb{E}[f(\overline{x_k})] \leq f(x^*) + \frac{\|x_0 - x^*\|_2^2}{2tk} + \frac{t\sigma^2}{2}$
KNOWN DATA
UNKNOWN DATA
TO LABEL

FD 1: re-considering the current compute and memory assumptions

FD 2: better theoretic understanding of the limits and guarantees in CL

FD 3: towards large scale continual learning and allowing small updates in large models.

FD 4: integrating the labeling process and data acquiring steps into the CL-algorithm

Figure 2: An overview of the future directions (FD) discussed in Section 4

fashion compared to ingesting them in fixed but random order. Not having theoretical guarantees means that continual learning research is often shooting in the dark, hoping to solve a problem that we do not know is solvable in the first place.

To understand when and under which assumptions we can find solutions, new concepts in a number of directions need to be developed in order to theoretically grasp continual learning in its full breadth. A key aspect is optimization. In which sense and under which conditions do continual learning algorithms converge to stable solutions? And what kind of generalization can we expect? We want to emphasize that we should not be misguided by classical notions of those concepts. It might be, for instance, more insightful to think of continual learning as tracking a time-varying target when reasoning about convergence (e.g. Abel et al., 2023), and classic, static, notions of generalization might not work here, although initial results by Zimin & Lampert (2019) are promising. Even if it is possible to find a good solution, it is unclear whether this is achievable in reasonable time, and crucially, whether it can be more efficient than re-training from scratch. Knoblauch et al. (2020) show that even in ideal settings continual learning is NP-complete, yet Mirzadeh et al. (2021) empirically illustrate that often there are linear low-loss paths to the solution, reassuring that solutions that are easy to find are not unlikely to exist.

Not all continual learning is equally difficult. An important factor is the relatedness of old and new data. In domain adaptation, David et al. (2010) have shown that without assumptions on the data, some adaptation tasks are simply impossible. Empirically (Zamir et al., 2018) and to some extent theoretically (Prado & Riddle, 2022), we know that in many cases transfer is successful because most tasks are related. Similar results for continual learning are scarce. Besides data similarity, the problem setting is an important second facet. For instance, class incremental learning is much harder than its task-incremental counterpart, as it additionally requires the predictions of task-identities (van de Ven et al., 2022; Kim et al., 2022). We believe that understanding the *difficulty of a problem* and having *formal tools expressive enough to describe or understand relatedness between natural data* will allow a more principled approach, and better guarantees on possible results.

Finally, theory in continual learning might simply be necessary to deploy continual learning models in a trustworthy manner. It requires models to be certified (Huang et al., 2020), i.e. they need to be thoroughly tested before deployment to work as intended. It is however unclear how this would fare in a continual learning setting, as by design, such models will be updated after deployment.

4.3 Large scale continual learning

Most of the problems in Section 3 start when there is a change in the environment of the model and it needs to be updated. These changes are often are small compared to the preceding training. The initial models, often referred to as foundation models, are typically powerful generalist models that can perform well on various downstream tasks, e.g. Oquab et al. (2023); Radford et al. (2021). However, performance gains are generally seen when adapting these models to specific tasks or environments, which compromises the initial knowledge in the pretrained model. In a continuously evolving world, one would expect that this knowledge is subject to be continuous editing, updating, and expansion, without losses in performance. When investigating continual learning that starts with large-scale pretrained models, the challenges might differ from those encountered in continual learning from random initializations and smaller models.

In contrast to smaller models, the required adjustments to accommodate new tasks are usually limited compared to the initial training phase, which may result in forgetting being less pronounced than previously anticipated. It is an open questions which continual learning techniques are more effective in such a case. For example, Xiang et al. (2023) suggest that parameter regularization mechanisms (Kirkpatrick et al., 2017) are more effective than functional regularization (e.g. distillation approaches (Li & Hoiem, 2017)) in reducing forgetting in a large language model. Additionally, it might not be necessary to update all parameters, which in itself is computationally expensive for large models. Approaches considering adapters (Houlsby et al., 2019; Jia et al., 2022; Li & Liang, 2021), low rank updates (Hu et al., 2022a) or prompting (Jung et al., 2023), are argued to be more feasible in this setting. Freezing, or using non-uniform learning rates, might also be necessary when data is limited to prevent optimization and overfitting issues. How to adapt models if the required changes are comparatively small to the original training remains an interesting research direction, with promising initial results (Wang et al., 2022c; Li et al., 2023; Panos et al., 2023).

Lastly, in the large scale learning setting there is a paradigm shift from end-to-end learning towards more modular approaches, where different components are first trained and then stitched together. It is somewhat of an open question of what implication this has for continual learning (Ostapenko et al., 2022; Cossu et al., 2022). In the simplest scenario, one could decouple the learning of a representation, done with e.g. contrastive unsupervised learning, versus that of classifier with supervision (e.g. Alayrac et al., 2022). Yet this idea can be extended towards using multiple (e.g. domain specific) experts (Ramesh & Chaudhari, 2021) and using more than one modality (e.g. vision and speech) (Radford et al., 2021). A better understanding of how continual learning algorithms can exploit these setting is required to expand beyond the end-to-end paradigms currently used.

While there has been promising research in these directions, we believe that considerably more is needed. So far, we do not have a strong understanding of the possibilities and limits of small updates on large pre-trained models, and how the training dynamics are different than the smaller-scale models typically used in continual learning. Further research in the relation between new data and pre-training data might open up new opportunities to more effectively apply these smaller updates, and will ultimately make continual learning more effective in handling all sorts of changes in data distributions. Understanding the interplay between memory and learning, and how to exploit the modular structure of this large model could enable specific ways to address the continual learning problem.

4.4 Continual learning in a real-world environment

Continual learning, in its predominant form, is centered around effective and resource-efficient accumulation of knowledge. The problem description typically starts whenever there is some form of new data available, see Section 3. How the data is produced is a question that is much less considered in continual learning. We want to emphasize that there is a considerable overlap between machine learning subfields (Mundt et al., 2022), in particular in those that are concerned with both detecting change in data distributions and techniques that reduce the required effort in labeling data. It will be important to develop continual learning algorithms with these in mind. These fields depend on and need each other to solve real-world problems, making it crucial that their desiderata align.

Open world learning (Bendale & Boult, 2015) is such a closely related field. Early work on open-world learning focused on detecting novel classes of objects that were not seen during training, relaxing the typical closed-world assumption in machine learning. A first step to realize open-world learning is detecting a change in incoming data, more formally known as out-of-distribution (OOD) or novelty detection. Detecting such changes requires a certain level of uncertainty-awareness of a model, i.e. it should quantify what it does and does not know. This uncertainty can be split into aleatoric uncertainty, which is an irreducible property of the data itself, and epistemic uncertainty, a result of the model not having learned enough (Hüllermeier & Waegeman, 2021). When modeled right, the latter can provide a valuable signal to identify what should be changed in continually trained models (Ebrahimi et al., 2020). Alternatively, it provides a theoretically grounded way for active learning, which studies how to select the most efficient unlabeled data points for labeling (Settles, 2009; Nguyen et al., 2022).

Even when OOD data is properly detected, it might not be directly usable. It can be unlabeled, without sufficient meta-data, or in the worst case corrupted. Many CL algorithms require the new data to be labeled before training, which is always costly and often difficult in e.g. on-device applications. This process makes it likely that when solving problems as described in Section 3, a model has access to a set of unlabeled data, possibly extended by some labeled samples that are obtained using active learning techniques. To successfully work in such an environment, a model should be able to update itself in a self- or semi-supervised way, an idea recently explored in Fini et al. (2022).

Continual learning depends on the data available to update a model. It is thus important to develop CL algorithms that are well calibrated, capable of OOD detection and learning in an open world. Further, in many settings (see Section 3.3), new data will not, or only partly, be labeled, which requires semi- or self-supervised continual learning (Mundt et al., 2023). We recommend working towards future continual learning algorithms with these considerations in mind, as methods that rely less on the fully labeled, closed-world assumption will likely be more practically usable in the future.

5 Conclusion

In this work, we first surveyed the current continual learning field, and showed that many papers study the memory-restricted setting with little or no concern for the computational cost. The problems we introduced all require some form of continual learning, not because it is a nice-to-have, but because the solution inherently depends on continual learning. Finally, we established four research directions in continual learning that we find promising, in the light of the scenarios we described. In summary, many of these applications are more compute-restricted than memory-restricted, so we vouch for exploring this setting more. Further, we believe a better theoretical understanding, a larger focus on pre-training and comparatively small future updates, and greater attention to how data is attained, will help us solving these problems, and make continual learning a practically useful tool to solve the described and other machine learning problems.

Broader impact

This paper does not present any new algorithm or dataset, hence the potential *direct* societal and ethical implications are rather limited. However, continual learning and applications thereof, as we have examined, may have a long-term impact. Reducing computational cost can positively affect the environmental impact machine learning has. Easily editable networks, or ways to quickly update parts of networks as discussed in Section 3.1 and 3.4, may further democratize the training of machine learning model. Yet this also means that it can be exploited by malicious actors to purposely inject false information in a network. Predictions made by those networks could misinform people or lead to harmful decisions. Excessive personalization as described in Section 3.2 may negatively impact community solidarity, yet benefit the individual.

References

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. *arXiv preprint arXiv:2307.11046*, 2023.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

Sridhar Alla, Suman Kalyan Adari, Sridhar Alla, and Suman Kalyan Adari. What is MLOPs? *Beginning MLOps with MLFlow: Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure*, pp. 79–124, 2021.

Dario Amodei and Danny Hernandez. AI and compute. `https://openai.com/research/ai-and-compute`, 2018. Online; accessed 20-June-2023.

Jordan Ash and Ryan P Adams. On warm-starting neural network training. *Advances in Neural Information Processing Systems*, 33:3884–3894, 2020.

Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1893–1902, 2015.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual International Conference on Machine Learning*, pp. 41–48, 2009.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Tudor Berariu, Wojciech Czarnecki, Soham De, Jorg Bornschein, Samuel Smith, Razvan Pascanu, and Claudia Clopath. A study on the plasticity of neural networks. *arXiv preprint arXiv:2106.00042*, 2021.

Vivek Chavan, Paul Koch, Marian Schlüter, and Clemens Briese. Towards realistic evaluation of industrial continual learning scenarios with an emphasis on energy consumption and computational footprint. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11506–11518, 2023.

Niv Cohen, Rinon Gal, Eli A Meirom, Gal Chechik, and Yuval Atzmon. "this is my unicorn, fluffy": Personalizing frozen vision-language representations. In *European Conference on Computer Vision*, pp. 558–577. Springer, 2022.

Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. Continual pre-training mitigates forgetting in language and vision. *arXiv preprint arXiv:2205.09357*, 2022.

William Dally. On the model of computation: point. *Communications of the ACM*, 65(9):30–32, 2022.

Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 129–136. JMLR Workshop and Conference Proceedings, 2010.

Saket Dingliwal, Monica Sunkara, Srikanth Ronanki, Jeff Farris, Katrin Kirchhoff, and Sravan Bodapati. Personalization of CTC speech recognition models. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 302–309. IEEE, 2023.

Shibhansh Dohare, Juan Hernandez-Garcia, Parash Rahman, Richard Sutton, and Rupam Mahmood. Loss of plasticity in deep continual learning. *Research Square preprint PPR: PPR727015*, 2023. doi: 10.21203/rs.3.rs-3256479/v1.

Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. Uncertainty-guided continual learning with bayesian neural networks. *International Conference on Learning Representations*, 2020.

Enrico Fini, Victor G Turrisi Da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2022.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999.

Haotian Fu, Shangqun Yu, Michael Littman, and George Konidaris. Model-based lifelong reinforcement learning with bayesian exploration. *Advances in Neural Information Processing Systems*, 35:32369–32382, 2022.

Yasir Ghunaim, Adel Bibi, Kumail Alhamoud, Motasem Alfarra, Hasan Abed Al Kader Hammoud, Ameya Prabhu, Philip HS Torr, and Bernard Ghanem. 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.

Eileen Guo. A roomba recorded a woman on the toilet. how did screenshots end up on facebook? `https://www.technologyreview.com/2022/12/19/1065306/roomba-irobot-robot-vacuums-artificial-intelligence-training-data-privacy/`, 2022. Online; accessed 11-October-2023.

Guy Hacohen, Leshem Choshen, and Daphna Weinshall. Let's agree to agree: Neural networks share classification order on real datasets. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3950–3960. PMLR, 2020.

Raia Hadsell, Dushyant Rao, Andrei A Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, 24(12):1028–1040, 2020.

Niv Haim, Gal Vardi, Gilad Yehudai, Ohad Shamir, and Michal Irani. Reconstructing training data from trained neural networks. *Advances in Neural Information Processing Systems*, 35:22911–22924, 2022.

Md Yousuf Harun, Jhair Gallardo, Tyler L. Hayes, and Christopher Kanan. How efficient are today's continual learning algorithms? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2431–2436, June 2023a.

Md Yousuf Harun, Jhair Gallardo, Tyler L Hayes, Ronald Kemker, and Christopher Kanan. SIESTA: Efficient online continual learning with sleep. *Transactions on Machine Learning Research*, 2023b.

Md Yousuf Harun, Jhair Gallardo, and Christopher Kanan. GRASP: A rehearsal policy for efficient online continual learning. *arXiv preprint arXiv:2308.13646*, 2023c.

Tyler L Hayes and Christopher Kanan. Online continual learning for embedded devices. In *Conference on Lifelong Learning Agents*, 2022.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022a. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

Hexiang Hu, Ozan Sener, Fei Sha, and Vladlen Koltun. Drinking from a firehose: Continual learning with web-scale natural language. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5): 5684–5696, 2022b.

Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37:100270, 2020.

E. Hüllermeier and W. Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021. doi: 10.1007/s10994-021-05946-3.

Chip Huyen. Real-time machine learning: challenges and solutions, Jan 2022. URL `https://huyenchip.com/2022/01/02/real-time-machine-learning-challenges-and-solutions.html#towards-continual-learning`. Online; accessed 14-November-2023.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pp. 709–727. Springer, 2022.

Dahuin Jung, Dongyoon Han, Jihwan Bang, and Hwanjun Song. Generating instance-level prompts for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11847–11857, 2023.

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. Achieving forgetting prevention and knowledge transfer in continual learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Zixuan Ke, Yijia Shao, Haowei Lin, Hu Xu, Lei Shu, and Bing Liu. Adapting a language model while preserving its general knowledge. In *Proceedings of The 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP-2022)*, 2022.

Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. A theoretical study on solving continual learning. In *Advances in Neural Information Processing Systems*, 2022.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Jeremias Knoblauch, Hisham Husain, and Tom Diethe. Optimal continual learning has perfect memory and is np-hard. In *International Conference on Machine Learning*, pp. 5327–5337. PMLR, 2020.

Akinwande Komolafe. Retraining model during deployment: Continuous training and continuous testing, 2023. URL `https://neptune.ai/blog/retraining-model-during-deployment-continuous-training-continuous-testing`. Online; accessed 30-June-2023.

Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. Machine learning operations (MLOPS): Overview, definition, and architecture. *IEEE Access*, 2023.

Dhireesha Kudithipudi, Mario Aguilar-Simon, Jonathan Babb, Maxim Bazhenov, Douglas Blackiston, Josh Bongard, Andrew P Brna, Suraj Chakravarthi Raja, Nick Cheney, Jeff Clune, et al. Biological underpinnings for lifelong learning machines. *Nature Machine Intelligence*, 4(3):196–210, 2022.

Dhireesha Kudithipudi, Anurag Daram, Abdullah Zyarah, Fatima tuz Zohora, James B. Aimone, Angel Yanguas-Gil, Nicholas Soures, Emre Neftci, Matthew Mattina, Vincenzo Lomonaco, Clare D. Thiem, and Benjamin Epstein. Uncovering design principles for lifelong learning ai accelerators. *Nature Electronics (Final Revisions)*, 2023.

Saurabh Kumar, Henrik Marklund, Ashish Rao, Yifan Zhu, Hong Jun Jeon, Yueyang Liu, and Benjamin Van Roy. Continual learning as computationally constrained reinforcement learning. *arXiv preprint arXiv:2307.04345*, 2023.

Lilly Kumari, Shengjie Wang, Tianyi Zhou, and Jeff A Bilmes. Retrospective adversarial replay for continual learning. *Advances in Neural Information Processing Systems*, 35:28530–28544, 2022.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1096, 2019.

Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34:29348–29363, 2021.

Sergey Levine, Aviral Kumar, George Tucker, and Justin fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Zhuowei Li, Long Zhao, Zizhao Zhang, Han Zhang, Di Liu, Ting Liu, and Dimitris N Metaxas. Steering prototype with prompt-tuning for rehearsal-free continual learning. *arXiv preprint arXiv:2303.09447*, 2023.

Junfan Lin, Zhongzhan Huang, Keze Wang, Xiaodan Liang, Weiwei Chen, and Liang Lin. Continuous transition: Improving sample efficiency for continuous control problems via mixup. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9490–9497. IEEE, 2021.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8:293–321, 1992.

Clare Lyle, Mark Rowland, Will Dabney, Marta Kwiatkowska, and Yarin Gal. Learning dynamics and generalization in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 14560–14581. PMLR, 2022.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Fmg_fQYUejf`.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=0DcZxeWfOPt`.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Martin Mundt, Steven Lang, Quentin Delfosse, and Kristian Kersting. CLEVA-compass: A continual learning evaluation assessment compass to promote research transparency and comparability. *International Conference on Learning Representations*, 2022.

Martin Mundt, Yongwon Hong, Iuliia Pliushch, and Visvanathan Ramesh. A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning. *Neural Networks*, 160:306–336, 2023.

V.L. Nguyen, M.H. Shaker, and E. Hüllermeier. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111(1):89–122, 2022. doi: 10.1007/s10994-021-06003-9.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Oleksiy Ostapenko, Timothee Lesort, Pau Rodríguez, Md Rifat Arefin, Arthur Douillard, Irina Rish, and Laurent Charlin. Foundational models for continual learning: An empirical study of latent replay, 2022. URL `https://arxiv.org/abs/2205.00329`.

Aristeidis Panos, Yuriko Kobe, Daniel Olmeda Reino, Rahaf Aljundi, and Richard E Turner. First session adaptation: A strong replay-free baseline for class-incremental learning. *arXiv preprint arXiv:2303.13199*, 2023.

Anastasia Pentina and Christoph H. Lampert. A PAC-bayesian bound for lifelong learning. In *ICML*, 2014.

Iuliia Pliushch, Martin Mundt, Nicolas Lupp, and Visvanathan Ramesh. When Deep Classifiers Agree: Analyzing Correlations Between Learning Order and Image Statistics. *European Conference on Computer Vision (ECCV)*, pp. 397–413, 2022.

Ameya Prabhu, Hasan Abed Al Kader Hammoud, Puneet K Dokania, Philip HS Torr, Ser-Nam Lim, Bernard Ghanem, and Adel Bibi. Computationally budgeted continual learning: What does matter? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3698–3707, 2023a.

Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, and Ozan Sener. Online continual learning without the storage constraint. *arXiv preprint arXiv:2305.09253*, 2023b.

Diana Benavides Prado and Patricia Riddle. A theory for knowledge transfer in continual learning. In *Conference on Lifelong Learning Agents*, 2022.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *International Conference on Machine Learning*, pp. 2827–2836. PMLR, 2017.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Rahul Ramesh and Pratik Chaudhari. Model zoo: A growing" brain" that learns continually. *arXiv preprint arXiv:2106.03027*, 2021.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Khalid Salama, Jarek Kazmierczak, and Donna Schut. Practitioners guide to MLOPS: A framework for continuous delivery and automation of machine learning. *Google Could White paper*, 2021.

Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems*, 34:23359–23373, 2021.

Tom Schaul, John Quan andIoannis Antonoglou, and David Silver. Prioritized experience replay. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green AI. *Communications of the ACM*, 63 (12):54–63, 2020.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* The MIT Press, second edition, 2018. URL `http://incompleteideas.net/book/the-book-2nd.html`.

European Union. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). *Official Journal L110*, 59:1–88, 2016.

Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022.

Liyuan Wang, Xingxing Zhang, Qian Li, Jun Zhu, and Yi Zhong. Coscl: Cooperation of small continual learners is stronger than a big one. In *European Conference on Computer Vision*, pp. 254–271. Springer, 2022a.

Zifeng Wang, Zheng Zhan, Yifan Gong, Geng Yuan, Wei Niu, Tong Jian, Bin Ren, Stratis Ioannidis, Yanzhi Wang, and Jennifer Dy. SparCL: Sparse continual learning on the edge. *Advances in Neural Information Processing Systems*, 35:20366–20380, 2022b.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022c.

Maciej Wołczyk, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual world: A robotic benchmark for continual reinforcement learning. *Advances in Neural Information Processing Systems*, 34:28496–28510, 2021.

Maciej Wolczyk, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Disentangling transfer in continual reinforcement learning. *Advances in Neural Information Processing Systems*, 35:6304–6317, 2022.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7971, 2022.

Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. Language models meet world models: Embodied experiences enhance language models. *arXiv preprint arXiv:2305.10626*, 2023.

Shipeng Yan, Jiangwei Xie, and Xuming He. 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.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 3712–3722, 2018.

Zeyu Zhao. The application of the right to be forgotten in the machine learning context: From the perspective of european laws. *Cath. UJL & Tech*, 31:73, 2022.

Alexander Zimin and Christoph H. Lampert. Tasks without borders: A new approach to online multi-task learning. In *ICML Workshop on Adaptive & Multitask Learning*, 2019. URL https://openreview.net/forum?id=HkllV5Bs24.

A Survey details

To verify the keywords *'incremental'*, *'continual'*, *'forgetting'*, *'lifelong'* and *'catastrophic'*, used to filter the papers based on their titles, we tested them using a manually collected validation set of which we are certain that they are continual learning related. This set was manually collected while doing research on continual learning over the past few years. The keywords were present in 96% of the paper titles. From each conference, we randomly picked 20 out of all matched papers, disregarding false positives.

It is common for to evaluate new methods and analyses on more than one benchmark. Often this means that the percentage of stored samples is not uniform across the experiments in a paper. In Figure 1, we showed the minimum percentage used, in Figure 3 we show the maximum. The conclusion remains the same, and the amount of stored samples is constrained in all but two benchmarks.

In Table 1 we provide a table of all the papers we used in the survey of Section 2, showing their minimal and maximal sample store ratio (SSR) i.e. the percentage of samples stored, as well as possibly other memory consumption. The last column mentions how they approached the computational cost.

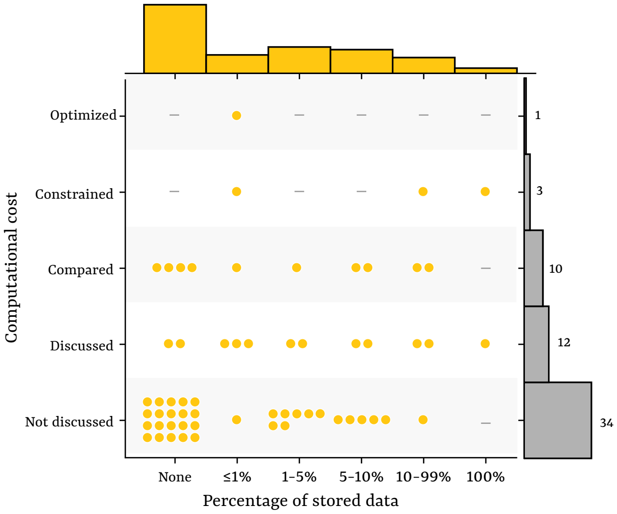


Figure 3: **Most papers strongly restrict memory use and do not discuss computational cost**. This figure is an alternate version of Figure 1, with the maximum percentage of stored samples rather than the minimum. Each dot represents one paper, illustrating what percentage of data their methods store (horizontal axis) and how computational complexity is handled (vertical axis). The majority of surveyed papers are in the lower-left corner: those that strongly restrict memory use and do not quantitatively approach computational cost (i.e. it is at most discussed). For more details, see Appendix.

Table 1: All papers used in the survey of Section 2. SSR refers to the sample store ratio, i.e. how much samples are stored in relation to the entire dataset.

#	Conference	Title	SSR (min)	SSR (max)	Memory (other)	Compute
1	ECCV	Balancing Stability And Plasticity Through Advanced Null Space In Continual Learning	0	0	null spaces	compared
2	ECCV	Class-Incremental Novel Class Discovery	0	0	prototypes	not discussed
3	ECCV	Prototype-Guided Continual Adaptation For Class-Incremental Unsupervised Domain Adaptation	0	0	prototypes	not discussed
4	ECCV	Few-Shot Class-Incremental Learning Via Entropy-Regularized Data-Free Replay	0	0	generators	not discussed
5	ECCV	Anti-Retroactive Interference For Lifelong Learning	0.02	0.2	/	discussed
6	ECCV	Long-Tailed Class Incremental Learning	0.01	0.04	/	not discussed
7	ECCV	Dlcft: Deep Linear Continual Fine-Tuning For General Incremental Learning	0.001	0.04	/	not discussed
8	ECCV	Generative Negative Text Replay For Continual Vision-Language Pretraining	0	0	/	not discussed
9	ECCV	Online Continual Learning With Contrastive Vision Transformer	0.001	0.02	/	not discussed
10	ECCV	Coscl: Cooperation Of Small Continual Learners Is Stronger Than A Big One	0	0.04	/	not discussed
11	ECCV	R-Dfcil: Relation-Guided Representation Learning For Data-Free Class Incremental Learning	0	0	generators	not discussed
12	ECCV	Continual Semantic Segmentation Via Structure Preserving And Projected Feature Alignment	0	0	/	discussed
13	ECCV	Balancing Between Forgetting And Acquisition In Incremental Subpopulation Learning	0	0	/	not discussed
14	ECCV	Few-Shot Class-Incremental Learning For 3d Point Cloud Objects	0.001	0.001	prototypes	not discussed
15	ECCV	Meta-Learning With Less Forgetting On Large-Scale Non-Stationary Task Distributions	0.002	0.002	/	compared
16	ECCV	Novel Class Discovery Without Forgetting	0	0	prototypes	not discussed
17	ECCV	Rbc: Rectifying The Biased Context In Continual Semantic Segmentation	0	0	model copies	not discussed
18	ECCV	Coarse-To-Fine Incremental Few-Shot Learning	0	0	/	not discussed
19	ECCV	Continual Variational Autoencoder Learning Via Online Cooperative Memorization	0.02	0.1	/	discussed
20	ECCV	Dualprompt: Complementary Prompting For Rehearsal-Free Continual Learning	0	0	prompts	not discussed
21	CVPR	Incrementer: Transformer For Class-Incremental Semantic Segmentation With Knowledge Distillation Focusing On Old Class	0	0	model copies	not discussed
22	CVPR	Real-Time Evaluation In Online Continual Learning: A New Hope	0.001	0.001	/	constrained
23	CVPR	Heterogeneous Continual Learning	0	0.004	generators	discussed
24	CVPR	Decoupling Learning And Remembering: A Bilevel Memory Framework With Knowledge Projection For Task-Incremental Learning	0	0	model copies	compared
25	CVPR	Geometry And Uncertainty-Aware 3d Point Cloud Class-Incremental Semantic Segmentation	0	0	model copies	not discussed
26	CVPR	Continual Detection Transformer For Incremental Object Detection	0.1	0.1	model copies	not discussed
27	CVPR	Continual Semantic Segmentation With Automatic Memory Sample Selection	0.001	0.01	/	discussed
28	CVPR	Adaptive Plasticity Improvement For Continual Learning	0	0	gradient bases	compared
29	CVPR	Vqacl: A Novel Visual Question Answering Continual Learning Setting	0.001	0.09	/	not discussed
30	CVPR	Task Difficulty Aware Parameter Allocation & Regularization For Lifelong Learning	0	0	model copies	discussed
31	CVPR	Computationally Budgeted Continual Learning: What Does Matter?	1	1	/	constrained
32	CVPR	Comformer: Continual Learning In Semantic And Panoptic Segmentation	0	0	model copies	not discussed
33	CVPR	Pivot: Prompting For Video Continual Learning	0.006	0.1	/	not discussed
34	CVPR	Class-Incremental Exemplar Compression For Class-Incremental Learning	0.003	0.02	/	discussed
35	CVPR	Pcr: Proxy-Based Contrastive Replay For Online Class-Incremental Continual Learning	0.002	0.1	/	not discussed
36	CVPR	Attriclip: A Non-Incremental Learner For Incremental Knowledge Learning	0	0	/	not discussed
37	CVPR	Learning With Fantasy: Semantic-Aware Virtual Contrastive Constraint For Few-Shot Class-Incremental Learning	0	0	prototypes	not discussed
38	CVPR	On The Stability-Plasticity Dilemma Of Class-Incremental Learning	0.01	0.01	/	discussed
39	CVPR	Metamix: Towards Corruption-Robust Continual Learning With Temporally Self-Adaptive Data Transformation	0.01	0.06	/	compared
40	CVPR	Exploring Data Geometry For Continual Learning	0.004	0.04	/	not discussed
41	NeurIPS	Uncertainty-Aware Hierarchical Refinement For Incremental Implicitly-Refined Classification	0	0	model copies	not discussed
42	NeurIPS	Learning A Condensed Frame For Memory-Efficient Video Class-Incremental Learning	0.001	0.03	/	not discussed
43	NeurIPS	S-Prompts Learning With Pre-Trained Transformers: An Occam's Razor For Domain Incremental Learning	0	0	/	not discussed
44	NeurIPS	Note: Robust Continual Test-Time Adaptation Against Temporal Correlation	0.5	0.5	/	discussed
45	NeurIPS	Decomposed Knowledge Distillation For Class-Incremental Semantic Segmentation	0.008	0.1	/	discussed
46	NeurIPS	Few-Shot Continual Active Learning By A Robot	0	0	prototypes	not discussed
47	NeurIPS	Navigating Memory Construction By Global Pseudo-Task Simulation For Continual Learning	0.004	0.04	/	compared
48	NeurIPS	Sparcl: Sparse Continual Learning On The Edge	0.004	0.01	/	optimized
49	NeurIPS	A Simple But Strong Baseline For Online Continual Learning: Repeated Augmented Rehearsal	0.01	0.1	/	compared
50	NeurIPS	Lifelong Neural Predictive Coding: Learning Cumulatively Online Without Forgetting	0	0	/	not discussed
51	NeurIPS	A Theoretical Study On Solving Continual Learning	0.004	0.04	/	discussed
52	NeurIPS	Beyond Not-Forgetting: Continual Learning With Backward Knowledge Transfer	0	0	gradient bases	compared
53	NeurIPS	Task-Free Continual Learning Via Online Discrepancy Distance Learning	0.04	0.2	/	compared
54	NeurIPS	Disentangling Transfer In Continual Reinforcement Learning	0.1	0.1	/	not discussed
55	NeurIPS	Less-Forgetting Multi-Lingual Fine-Tuning	0.5	0.5	/	not discussed
56	NeurIPS	Model-Based Lifelong Reinforcement Learning With Bayesian Exploration	1	1	/	discussed
57	NeurIPS	Alife: Adaptive Logit Regularizer And Feature Replay For Incremental Semantic Segmentation	0.01	0.04	/	not discussed
58	NeurIPS	Retrospective Adversarial Replay For Continual Learning	0.004	0.2	/	constrained
59	NeurIPS	Acil: Analytic Class-Incremental Learning With Absolute Memorization And Privacy Protection	0	0	/	not discussed
60	NeurIPS	Memory Efficient Continual Learning With Transformers	0.1	0.16	/	compared