

Beyond Supervised Continual Learning: a Review

Benedikt Bagus¹, Alexander Gepperth¹ and Timothée Lesort²

¹ University of Applied Sciences Fulda - Department of Computer Science
Leipzigerstraße 123, 36037 Fulda - Germany

² Montréal University (UdeM), Mila
6666 St-Urbain Street, Montréal, QC - Canada

Abstract. *Continual Learning* (CL, sometimes also termed *incremental learning*) is a flavor of machine learning where the usual assumption of stationary data distribution is relaxed or omitted. When naively applying, e.g., DNNs in CL problems, changes in the data distribution can cause the so-called *catastrophic forgetting* (CF) effect: an abrupt loss of previous knowledge. Although many significant contributions to enabling CL have been made in recent years, most works address supervised (classification) problems. This article reviews literature that study CL in other settings, such as learning with reduced supervision, fully unsupervised learning, and reinforcement learning. Besides proposing a simple schema for classifying CL approaches w.r.t. their level of autonomy and supervision, we discuss the specific challenges associated with each setting and the potential contributions to the field of CL in general.

1 Introduction

Continual learning is a field of machine learning where the data distribution is not static. It is a natural framework for many practical problems where the data arrives progressively, and the model learns continuously. For example, in robotics, robots need to adapt to their environments to interact and realize actions constantly, or recommendation systems also need to adapt constantly to the new content available and the new needs of users. However, in recent years, the field of continual learning has focused mainly on one type of classification scenario: class-incremental. This scenario evaluates how models can learn a class once and remember it when new class data arrives. While it is important to solve this problem, using only one type of scenario can lead to over-specialized solutions that can not generalize to different settings. In this paper, we propose to review the literature dealing with other settings than the default one (class-incremental) and more generally, fully supervised scenarios. The goal is to shed light on efforts made to diversify the evaluation of continual learning.

2 Framework and Goals of Continual Learning

Continual learning (CL) is a machine learning sub-field that studies learning under time-varying data distributions. This relaxes one of the fundamental assumptions of statistical learning theory ([1]), which states that the data follows

a stationary distribution. One advantage of this assumption is its simplicity, whereas CL scenarios are very diverse, depending on the nature of the non-stationarity. In CL, it is therefore very important to clearly define the scenario, the goals of learning, the evaluation measures, and the loss functions. The following section describes typical non-stationarities of the data distribution that have been considered in the literature (see also [2, 3]).

2.1 Data distribution Drifts

CL under data distribution shifts needs memorization mechanisms adapted for the type of non-stationarity, which requires assumptions by the used algorithms in the face of an infinite number of possible ways a distribution can be non-stationary. For simplicity, we will list several typical definitions used for supervised learning (e.g., classification) but that can be generalized to other settings.

A simple way to categorize non-stationarities is based on class information. We may distinguish two types of shifts [2] in this case: *concept shift*, where the annotation of existing data changes [4] and *virtual shift*, where we get new data, but the annotation does not change. Usually, the term *shift* is employed for sudden changes in data distributions, whereas the term *drift* is used for gradual changes over time. In supervised CL, virtual shifts are the most common non-stationarities that have been studied. We can distinguish the special cases of *virtual concept shift*, implying new data and new labels, and *domain shift*, where new data of known labels are observed. Those two settings are also known as respectively *class-incremental* and *instance-incremental* [5, 6]. The objectives to be optimized may change over time as well [3], as in continual RL [7, 8, 9].

2.2 Common CL constraints

If CL were not subject to constraints, there would be a simple solution to any scenario. It involves storing all incoming samples and re-training every time a decision is expected, [10]. This entails a time and memory complexity that is at least linear in the number of processed samples. However, most CL proposals assume that memory is limited in some way, preventing this (obvious) solution. Many approaches similarly assume that the storage of samples is restricted. Other resources constraints subject of study are: *computational cost*, *memory*, *data privacy*, *fast adaptation*, *inference speed*, *transfer*. Other constraints that are more related to the reliability of approaches are *stability* and *explainability*. A discussion of CL constraints can be found in, e.g., [11], whereas evaluation measures that take these constraints into account are given in [Lesort2019Continual].

3 The default scenario for CL

A particular supervised setting, which we will refer to as the *default scenario*, is currently dominating CL literature. It is based on a classification problem divided into a small number of *sub-tasks*. Virtual concept shift occurs abruptly at sub-task boundaries by the apparition of new samples belonging to previously

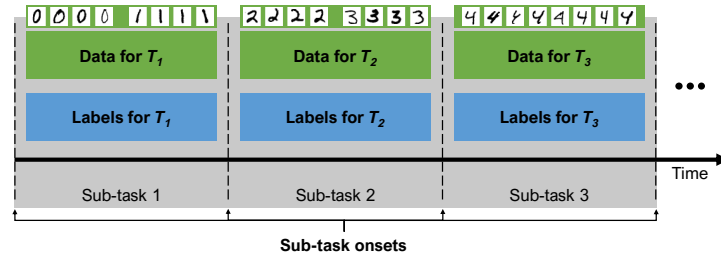


Figure 1: The default scenario for CL. The data stream is assumed to be partitioned into *sub-tasks*. Statistics within a sub-task are considered stationary, and sub-task data and labels (targets) are assumed to be disjoint, i.e., from different classes. Sub-task onsets are assumed to be known as well.

unseen classes, see Fig 1. Usually, the sub-task onsets/boundaries are known. A consequence of this disjointness of sub-tasks is that no pure concept shift is involved: the annotation for a given data point will never change or be subject to conflicts as learning progresses.

In the default scenario, the goal of CL is usually to learn the complete statistics of all sub-tasks as if they had been processed all at once, rather than one after the other. Samples may be processed one by one, or all at once for a given sub-task. In some works, the sub-task index is known at test time and used for selecting the correct head of a multi-headed DNN.

The assumptions made in the default scenario are justified in many cases. However, other scenarios, e.g., in robotics, may be found where they impose too severe restrictions. Moreover, many characteristics of the default scenario, such as “drift are abrupt” or, “tasks are not revisited” lend themselves to *benchmark overfitting*. Consider sub-task T_2 in Fig 1: here, a DNN could punish incorrect decisions for class “1” more strongly than incorrect decisions for classes from the current sub-task, since the default scenario assumes that sub-tasks are disjoint. Even if researchers do not consciously exploit these assumptions, the employed CL algorithms may rely on them indirectly. It is fundamental to perform experiments in scenarios where these assumptions do not hold.

Thus, creating diverse benchmarks, as well as approaches that do not critically rely on the assumptions from the default scenario, should be an ongoing effort. This effort should be pushed notably by existing continual learning libraries such as Continuum [12], Avalanche [13] or Sequoia [7].

3.1 CL approaches for the default scenario

This section is not meant to be exhaustive, since the default scenario is not the focus of this article, please see, e.g., [14, 15] for this.

Broad strategies for performing CL in the default scenario are regularization [16, 17], replay [18, 19, 20] and dynamic architectures [21, 22, 23, 24, 25]. Regularization penalizes changes to model parameters that are deemed important for

past sub-tasks. This is usually achieved by penalty terms in the loss function, and it is assumed that new sub-tasks add only new data and classes. Dynamic architecture methods extend models over time in order to separate previously learned parameters from currently optimized ones, thus reducing cross-talk and forgetting (but equally assuming that new sub-tasks contain only new data). Replay methods store received data for subsequent use in re-training (rehearsal). Instead of relying on stored data, re-training can also be performed using samples produced by generative models (generative replay).

Replay is known to be an effective method for preventing catastrophic forgetting (CF), especially in class-incremental settings [26, 14], but also for continual reinforcement learning [27, 28, 19] or unsupervised learning [29, 30, 31].

3.2 Metrics and evaluation procedures

In the default scenario, various measures related to the classification error are common, which have been discussed in, e.g., [32, 33, 34]. A common baseline is termed *cumulative performance*, obtained by evaluating models on the merged test sets from all sub-tasks, which corresponds to learning with stationary statistics. This baseline is often considered an upper bound for CL performance.

In addition, [17] proposed the notions of forward and backward transfer: forward transfer (FT) measures how training on sub-task i impacts performance on a future sub-task $j > i$. For backward transfer (BT), the impact on previous sub-tasks $j < i$ is considered. The common case in CL is negative BT indicating forgetting, but positive BT is theoretically possible as well.

Many authors, e.g., [16], assume that (although sub-tasks are presented sequentially) all sub-task data are available for model selection and hyperparameter tuning. For example, tuning EWC's regularization strengths λ_i for each sub-task is often done in hindsight.

Some authors [35, 36, 37], especially in works using multi-head DNNs, assume that the sub-task ID is known during testing. In the limit where each sub-task contains only a single class, providing the sub-task ID at test time means providing the class label. Even if sub-tasks are more diverse, the sub-task ID contains significant information and may thus confer unfair advantages. The question of evaluation protocols in CL is discussed in [11, 38].

3.3 Benchmarks

Benchmarks for the default CL scenario are mostly derived from datasets such as MNIST, CIFAR10/100, Imagenet, SVHN etc. to create *class-incremental* or *domain-incremental* scenarios. The *permuted MNIST* benchmark was initially popular [16, 39, 40] but is less so now because it can, to good accuracy, be solved even without dedicated CL schemes [11]. Some authors use Atari games [41], Mujoco [42] or Meta-World [43] as benchmarks. CL-specific variants of standard benchmarks such as, e.g., *colored MNIST* [44] are widely used as well since they can be used to investigate specific aspects of CL, see [45, 46].

4 Generalizations of the default scenario

The default scenario is useful for evaluation and represents a controlled setting for CL. In less controlled settings, fully annotated data may be unavailable, or supplementary constraints may be imposed. We find it convenient to introduce a new taxonomy of CL approaches based on their level of autonomy.

4.1 Classifying the autonomy level of CL algorithms

The various applications of continual learning can be grouped into autonomy levels. Obviously, CL should get harder as less and less human supervision is supplied. We identify two dimensions of autonomy:

Objective Autonomy denotes the autonomy regarding the objective to achieve (labels, targets, rewards), which we group into 4 levels:

- **Level 0:** Full data annotation: supervised.
- **Level 1:** Sparse labelization: RL, active learning, sparse training.
- **Level 2:** No annotation for training, query for fast adaptation.
- **Level 3:** No annotation for training, zero-shot adaptation.

For levels 2 and 3, continual training can be seen as pre-training for the unknown future task. If the scenario objective is unsupervised, then we can assume that it is similar to full data annotation.

Continual Learning Autonomy concerns autonomy regarding distribution shifts (task label, task boundaries):

- **Level 0:** Full task annotation at train and test.
- **Level 1:** Full task annotation at train, no test task labels.
- **Level 2:** Sparse task annotation at train, no test task labels: example task boundaries only without train task label.
- **Level 3:** No task labels at all: task-agnostic.

This taxonomy still holds for smooth transitions. For class-incremental problems, the task-agnostic setting is inapplicable since the task information is in the class labels. We can now characterize CL by (i, j) , where i represents objective autonomy, and j for the CL autonomy. Hence, $(0, 1)$ describes a class-incremental scenario, i.e., a classification setting with fully annotated data and task labels for training but not for test data. It is important to note that those ratings classify the complexity of a given scenario. For example, the default scenario (class-incremental) assesses a complexity level of $(0, 1)$, domain incremental without task labels would be $(0, 2)$, task agnostic continual RL [47] $(1, 3)$. To validate the autonomy of an approach, it should be evaluated in adequate scenarios.

Investigating the cost of lowering or increasing a task's complexity level is fundamental for applications of CL. We want to make our algorithms scale up to arbitrary complexity levels, but in practice, we would always choose the lowest possible complexity. If possible, solving a scenario with a complexity level $(0, 1)$ is more efficient than solving the same scenario with a level of $(3, 3)$.

4.2 Towards a Generalization of the Default Setting

Some variants of supervised CL exist that alleviate the need for annotated data. The reduction of annotations can apply for restricted access to the task labels as in task-agnostic CL [48, 49], or by reducing the labels' availability as in continual active learning [50, 51], or in semi-supervised continual learning, [52]. As described in Sec. 4.1, reducing access to task supervision lead to evaluating a CL autonomy level of 2, and removing all access to it lead to evaluating a CL autonomy level of 3. On the other hand, reducing data annotation lead to an objective autonomy of 1 instead of 0 when the full annotation is available.

Among potential supplementary constraints, data can be streamed without the possibility of multi-epoch training as in online training [53, 54], or the data can be imbalanced [55, 56], or mixed with spurious features [57]. Scenarios where the annotations change over time (real concept drift) have been investigated in [3, 58, 4]. Some contributions [59] relax the condition of disjoint sub-tasks and assess the impact of several fundamental CL strategies such as regularization and replay. Yet others [60, 61] demonstrate that detecting sub-task boundaries autonomously is feasible by using density estimation methods.

To the generalized settings cited above, supplementary constraints (as discussed in Sec. 2.2) may be added, making them even harder. If this brings CL closer to real-life applications, solutions are required that do not overfit a particular CL setting. Currently, the field of CL is fundamentally meta: implicitly, the goal is not to train the best possible model w.r.t. non-continual baselines, but rather to create algorithms that show maximal generalization to other CL settings. Therefore, experimenting with generalized supervised scenarios can assess algorithms' robustness and improve generalized CL.

5 Unsupervised Continual Learning

Whether a machine learning task is considered supervised or not depends on the formulation of the loss function. In fact, no assumptions whatsoever are made concerning the loss in the definition of CL given in Sec. 1. Therefore, CL is naturally transferred to density modeling, clustering, generative learning, and unsupervised representation learning.

5.1 Density modeling

Density modeling aims at approximating the probability density of a given set of data samples directly by minimizing a log-likelihood loss. Typically, this is achieved using *mixture models*, which model the data density as a weighted sum of N parameterized component densities, e.g., multivariate Gaussian densities or Dirichlet distributions. Density modeling allows performing, among other functions, Bayesian inference and sampling. These functionalities spawned increased interest in mixture modeling a few years back, particularly in robotics [62, 63, 64, 65]. The main issue for CL in mixture modeling is that concept drift or shift may require an adaptation of N , which motivates heuristics for adding and

removing components. Current approaches using generative replay are proposed in, e.g., [30]. Mixture models usually adapt only a small subset of components for each update step due to their intrinsic reliance on distances instead of scalar products. This is why they are less prone to catastrophic forgetting than DNNs, an effect that has been demonstrated for self-organizing maps in [66] which are an approximation to Gaussian Mixture Models (GMMs), see [67]. Modeling the data density allows partitioning data space into Voronoi cells, in each of which a separate linear classifier model can be trained. This is the essence of the popular Locally Weighted Projection Regression (LWPR) algorithm [68] which was explicitly constructed for continual classification in robotics.

5.2 Clustering

Clustering is, in a certain sense, an approximation to density modeling, although inference is limited to determining the precise component a given data sample was generated from. Clustering methods are normally trained using a k-means type of algorithm. CL for clustering algorithms faces the same basic issue as in density modeling: a potentially variable number N of cluster centers during concept drift or shift. This has been demonstrated in, e.g., [69, 70, 71].

5.3 Generative learning

Generative learning aims to generate realistic samples (typically images) that are similar to a set of training data. Typical models are generative adversarial networks (GANs), variational auto-encoders (VAEs), PixelCNN, FLoW or GLoW, but many other variants have been proposed, see, e.g., [72] for a review. Training such generative models can be performed, e.g., in the CL default scenario introduced in Sec. 3 (apart from the supervision information), which leads to catastrophic forgetting (CF) without additional measures. Several of the approaches used in supervised learning have been successfully applied to training generative models: knowledge distillation [73], EWC [74] and replay [75, 76, 77, 78, 29]. To our knowledge, no generic approaches that are specific to generative learning have been proposed, apart perhaps [79] where it is proposed to learn specific transformations. This, however, is very specific to a particular kind of (image) data and would have to be adapted if other kinds of data were targeted.

5.4 Continual Representation Learning

Unsupervised representations learning for downstream applications is a common use case for unsupervised learning. It was a motivations for various types of auto-encoders and generative models in the early days of deep learning. In CL, unsupervised criterion to learn representations might be useful to avoid overfitting a specific task and improving performance on downstream tasks [80, 30, 31]. Unsupervised pre-training can also be useful for learning a general feature extractor that can be frozen for future tasks [28, 81, 82].

5.5 Challenges of unsupervised CL

Unsupervised learning offers general learning criteria that can avoid the over-specialization of supervised training and reduce forgetting. Nevertheless, unsupervised CL faces the same challenges as supervised CL, and the default scenario for supervised CL can be transferred. Moreover, in practice, unsupervised training tends to be more complex than supervised training, especially for generation and density modeling, since it is harder to model a distribution than to determine a separating hypersurface between classes in data space. With the added complexity of CL, unsupervised learning can be a formidable problem, especially w.r.t. model and hyper-parameter selection.

6 Continual Reinforcement Learning

In reinforcement learning (RL), an agent learns to interact with its environment by choosing a specific action for each state based on a reward signal. The (unknown) underlying process is formalized as a Markov decision process (MDP), where an optimal policy maximizes an expected reward. This scenario is inherently a CL setting, since the evolution of the policy throughout the learning process will automatically lead to a non-stationary data distribution. Changes in the environment or in the objective to fulfill can lead to additional non-stationarities. Hence, RL requires the ability to cope with non-stationary data. We will use the term *Continual Reinforcement Learning* (CRL) for denoting RL in settings that go beyond the usual assumptions of non-stationarity made in conventional RL.

6.1 Existing Approaches

The works presented in [83, 84] introduce the importance of CL at an early stage and especially investigate them in the context of reinforcement learning. More recent works, e.g., [85, 19] revisit this area and consider additional aspects, such as catastrophic forgetting. Some frameworks to guide future research have also been published [5, 8]. Both provide a comprehensive overview of the synergies between continual and reinforcement learning.

RL Approaches Experience replay [86, 19, 87] is the most common approach to counter non-stationarities in RL. Several variants are introduced, e.g., [88, 89, 90, 91, 92]. Continuous control, multi-task, and multi-goal are also research topics intersecting with continual reinforcement learning, but their scenarios are not always defined in a consistent fashion in the literature. In general, the goal is to enable transfer learning between policies, which, however, omits the capacity for forgetting or re-adaptation. Some works assume a static objective [93, 94, 95, 96, 97], others a static agent and/or environment [98, 99, 100] or none of both [101, 102, 103]. In contrast, multi-agent reinforcement learning is mostly related to some kind of joint training and is hence not related to CL.

For CRL, the agent needs to acquire new skills to handle time-varying conditions, such as changes in environment [104], observations or actions, and addi-

tionally must retain the old knowledge. A variety of approaches has been published, among which knowledge-based distillations [27, 28] and context-based decompositions [105, 106] are popular. Other works are concerned with the employed model [107, 108, 109, 110], off-policy algorithms [111], policy gradient [112] or a task-agnostic perspective [47]. Evaluations of known CL methods (e.g., GEM, A-GEM, and replay) are also applied in the RL domain [113, 9].

Benchmarks An overview of CRL environments can be found in [114]. Dedicated benchmarks which allow a systematical assessment are: *Meta-World* [43], *Continual World* [115] and *L2Explorer* [116].

6.2 Assumptions in CRL

Three assumptions are commonly made in CRL: Foremost, a decomposition into sub-tasks is assumed, even if their onset is unknown since most dedicated CL methods (see Sec. 3.1) assume the existence of distinct sub-tasks. Another assumption concerns samples, which are assumed to be non-contradictory within sub-tasks, meaning the assessment of rewards changes only between sub-tasks, if it changes at all. Finally, it is a common assumption that knowledge of sub-task boundaries is provided. Most existing works are using information about sub-task boundaries as if they were provided by an oracle, without the possibility to recognize or determine them autonomously.

6.3 Challenges

Environment-related On a short timescale, observations will always be non-stationary, even if the environment is. In addition, the environment itself can change gradually or abruptly (*environment drift/shift*). This results in novel states or transitions between them, resulting in an enlargement of the MDP.

Goal-related By maximizing the reward signal, the agent attains a defined objective. If the reward function changes, the agent experiences divergent information, leading to an inconsistent policy. In this setting, the definition of states, actions and transitions does not change, so the underlying MDP remains structurally intact. However, other rewards are assigned to previously learned mappings, enforcing changes of transition probabilities.

Agent-related The decreasing influence of exploration, regardless of whether off- or on-policy methods are used, temporarily creates a source of non-stationarity, resulting in a time-varying sampling of the state-action space even with a static policy and a static environment. Additionally, sensors or actuators may degrade or undergo deliberate manipulations.

Sub-task definition For scenarios where the environment changes in a discrete fashion, we can introduce the notion of sub-tasks as in the default scenario for supervised CL, see Sec. 3. Sub-task onsets can be unknown, and the detection of boundaries may be difficult if the drifts are gradual. Moreover, the number of sub-tasks can be high, up to a point where the concept of sub-tasks becomes questionable. Lastly, similar states and actions may appear in sub-tasks, but

with different rewards, so sub-tasks may not be disjoint or even contradictory, requiring re-learning, a concept absent from the default CL scenario.

7 Discussion - Conclusion

The field of CL has expanded rapidly in recent years, which is why many aspects of CL are still fluid and not subject to a common consensus among researchers. This is evidenced by a wide variety of assumptions, evaluation metrics, see Sec. 3.2 and constraints, see Sec. 2.2. The so-called default scenario, see Sec. 3, is the nearest thing to a commonly agreed scenario, yet many details fluctuate strongly between contributions. This leads to several interesting consequences and opportunities for further research:

CL comparability A direct consequence is the difficulty to directly compare results of different articles. This underscores the need, in CL more than in other domains of machine learning, to describe evaluation procedures and, where possible, make use of existing libraries (see Sec. 3 and 6.1) and evaluation procedures. Furthermore, CL is a multi-objective problem (Sec. 2.2) where the cumulative performance is important, but where other measures (Sec. 3.2) matter as well.

CL autonomy As explained in Sec. 4.1, CL approaches should also be evaluated based on the complexity and autonomy of the scenario they can generalize to, to prevent them from overfitting to a specific CL scenario or assumptions.

CL scalability An aspect of quantitative performance measures is scalability. Depending on a potential application context, CL, even in the default scenario, may be faced with a huge number of sub-tasks, each again containing enormous amounts of samples. If this were not the case, the cumulative baseline, or some variant of GDumb (see Sec. 3.2), would be a much less costly alternative to using dedicated CL methods. Time and memory complexity w.r.t. the number of sub-tasks should be included in all new CL works to ensure comparability.

CL generalization As was shown in Sec. 4, 5 and 6, the default CL scenario of Sec. 3 can be generalized in many ways. These chapters show that open issues remain, both technical and conceptual, when generalizing CL.

This review article attempts to give an overview of the current state of CL beyond the purely supervised default scenario, see Sec. 3. We describe the various complexification of the default scenario and the different learning paradigms, and propose a classification based on the autonomy characteristics of algorithms. We believe that attempts to generalize CL pose important questions about the fundamental assumptions behind CL. We thus encourage CL researchers to carefully reflect upon the implicit, hidden assumptions in each CL approach they are dealing with and whether they can (and should) be relaxed. In a still-fluid field such as CL, a continuous re-examination of assumptions may lead to new solutions that strongly contribute to the advancement of the field.

References

- [1] V. Vapnik. *The nature of statistical learning theory*. Springer, 1999.
- [2] A. Gepperth and B. Hammer. "Incremental learning algorithms and applications". In: *ESANN*. Bruges, Belgium, 2016.
- [3] T. Lesort, M. Caccia, et al. "Understanding Continual Learning Settings with Data Distribution Drift Analysis". In: *arXiv:2104.01678* (2021).
- [4] M. Caccia, P. Rodriguez, et al. "Online Fast Adaptation and Knowledge Accumulation: a New Approach to Continual Learning". In: *NeurIPS* (2020).
- [5] T. Lesort, V. Lomonaco, et al. "Continual Learning for Robotics: Definition, Framework, Learning Strategies, Opportunities and Challenges". In: *Information Fusion* (2019).
- [6] G. M. van de Ven and A. S. Tolias. "Three scenarios for continual learning". In: *arXiv:1904.07734* (2019).
- [7] F. Normandin, F. Golemo, et al. "Sequoia: A Software Framework to Unify Continual Learning Research". In: *arXiv:2108.01005* (2021).
- [8] K. Khatarpal, M. Riemer, et al. "Towards Continual Reinforcement Learning: A Review and Perspectives". In: *arXiv:2012.13490* (2020).
- [9] B. Bagus and A. Gepperth. "A Study of Continual Learning Methods for Q-Learning". In: *IJCNN*. 2022.
- [10] A. Prabhu, P. H. Torr, et al. "GDumb: A simple approach that questions our progress in continual learning". In: *ECCV*. 2020.
- [11] B. Pfülb and A. Gepperth. "A comprehensive, application-oriented study of catastrophic forgetting in DNNs". In: *ICLR*. 2019.
- [12] A. Douillard and T. Lesort. "Continuum: Simple Management of Complex Continual Learning Scenarios". In: *arXiv:2102.06253* (2021).
- [13] V. Lomonaco, L. Pellegrini, et al. "Avalanche: an End-to-End Library for Continual Learning". In: *GitHub repository* (2021).
- [14] M. D. Lange, R. Aljundi, et al. "Continual learning: A comparative study on how to defy forgetting in classification tasks". In: *arXiv:1909.08383* (2019).
- [15] E. Belouadah, A. Popescu, et al. "A comprehensive study of class incremental learning algorithms for visual tasks". In: *Neural Networks* (2021).
- [16] J. Kirkpatrick, R. Pascanu, et al. "Overcoming catastrophic forgetting in neural networks". In: *Proc. of the national academy of sciences* (2017).
- [17] D. Lopez-Paz and M. Ranzato. "Gradient episodic memory for continual learning". In: *NeurIPS*. 2017.
- [18] S.-A. Rebuffi, A. Kolesnikov, et al. "iCARL: Incremental classifier and representation learning". In: *CVPR*. 2017.
- [19] D. Rolnick, A. Ahuja, et al. "Experience replay for continual learning". In: *NeurIPS*. 2019.
- [20] H. Shin, J. K. Lee, et al. "Continual learning with deep generative replay". In: *NeurIPS*. 2017.
- [21] A. A. Rusu, N. C. Rabinowitz, et al. "Progressive Neural Networks". In: *arXiv:1606.04671* (2016).
- [22] C. Fernando, D. Banarse, et al. "PathNet: Evolution Channels Gradient Descent in Super Neural Networks". In: *CoRR* (2017).
- [23] T. Veniat, L. Denoyer, et al. "Efficient Continual Learning with Modular Networks and Task-Driven Priors". In: *ICLR*. 2021.
- [24] O. Ostapenko, P. Rodriguez, et al. "Continual Learning via Local Module Composition". In: *NeurIPS*. 2021.
- [25] J. A. Mendez, H. van Seijen, et al. "Modular Lifelong Reinforcement Learning via Neural Composition". In: *ICLR*. 2022.
- [26] T. Lesort, T. George, et al. "Continual Learning in Deep Networks: an Analysis of the Last Layer". In: *arXiv:2106.01834* (2021).
- [27] R. T. Kalifou, H. Caselles-Dupré, et al. "Continual reinforcement learning deployed in real-life using policy distillation and sim2real transfer". In: *ICML Workshop on Multi-Task and Lifelong Learning*. 2019.
- [28] R. Traoré, H. Caselles-Dupré, et al. "DisCoRL: Continual Reinforcement Learning via Policy Distillation". In: *arXiv:1907.05855* (2019).
- [29] T. Lesort, H. Caselles-Dupré, et al. "Generative Models from the perspective of Continual Learning". In: *IJCNN*. 2019.
- [30] D. Rao, F. Visin, et al. "Continual Unsupervised Representation Learning". In: *arXiv:1910.14481* (2019).
- [31] D. Madaan, J. Yoon, et al. "Representational continuity for unsupervised continual learning". In: *ICLR*. 2021.
- [32] R. Kemker, M. McClure, et al. "Measuring catastrophic forgetting in neural networks". In: *AAAI*. 2018.
- [33] N. Diaz-Rodriguez, V. Lomonaco, et al. "Don't forget, there is more than forgetting: new metrics for Continual Learning". In: *arXiv:1810.13166* (2018).
- [34] D. Maltoni and V. Lomonaco. "Continuous learning in single-incremental-task scenarios". In: *Neural Networks* (2019).
- [35] J. Serra, D. Suris, et al. "Overcoming Catastrophic Forgetting with Hard Attention to the Task". In: *ICML*. 2018.
- [36] T. Doan, M. Abbana Bannani, et al. "A Theoretical Analysis of Catastrophic Forgetting through the NTK Overlap Matrix". In: *AISTATS*. 2021.
- [37] M. Farajtabar, N. Azizan, et al. "Orthogonal gradient descent for continual learning". In: *AISTATS*. 2020.
- [38] B. Pfülb. "Continual Learning with Deep Learning Methods in an Application-Oriented Context". PhD thesis. University of Applied Sciences Fulda, Germany, 2022.
- [39] D. Lopez-Paz and M.-A. Ranzato. "Gradient Episodic Memory for Continual Learning". In: *NeurIPS*. 2017.
- [40] M. Wortsman, V. Ramanujan, et al. "Supermasks in Superposition". In: *NeurIPS*. 2020.
- [41] V. Mnih, K. Kavukcuoglu, et al. "Playing Atari with Deep Reinforcement Learning". In: *arXiv:1312.5602* (2013).
- [42] E. Todorov, T. Erez, et al. "MuJoCo: A physics engine for model-based control." In: *IROS*. 2012.
- [43] T. Yu, D. Quillen, et al. "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning". In: *Conference on Robot Learning*. PMLR, 2020.
- [44] B. Kim, H. Kim, et al. "Learning not to learn: Training deep neural networks with biased data". In: *CVPR*. 2019.
- [45] I. Gat, I. Schwartz, et al. "Removing bias in multi-modal classifiers: Regularization by maximizing functional entropies". In: *NeurIPS*. 2020.
- [46] Y. Li and N. Vasconcelos. "Repair: Removing representation bias by dataset resampling". In: *CVPR*. 2019.
- [47] M. Caccia, J. Mueller, et al. "Task-Agnostic Continual Reinforcement Learning: In Praise of a Simple Baseline". In: *arXiv:2205.14495* (2022).
- [48] C. Zeno, I. Golan, et al. "Task Agnostic Continual Learning Using Online Variational Bayes". In: *arXiv:1803.10123* (2018).
- [49] X. He, J. Sygnowski, et al. "Task Agnostic Continual Learning via Meta Learning". In: *ArXiv* (2019).
- [50] M. Mundt, Y. W. Hong, et al. "A Wholistic View of Continual Learning with Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning". In: *arXiv:2009.01797* (2020).
- [51] M. Perkonig, J. Hofmanninger, et al. "Continual active learning for efficient adaptation of machine learning models to changing image acquisition". In: *Int. Conf. on Information Processing in Medical Imaging*. 2021.
- [52] J. Smith, J. Balloch, et al. "Memory-efficient semi-supervised continual learning: The world is its own replay buffer". In: *IJCNN*. IEEE, 2021.
- [53] A. Chaudhry, M. Ranzato, et al. "Efficient Lifelong Learning with A-GEM". In: *ICLR*. 2019.
- [54] J. Schwarz, J. Luketina, et al. "Progress & compress: A scalable framework for continual learning". In: *ICML*. 2018.
- [55] C. D. Kim, J. Jeong, et al. "Imbalanced Continual Learning with Partitioning Reservoir Sampling". In: *ECCV*. 2020.
- [56] A. Chrysakis and M.-F. Moens. "Online continual learning from imbalanced data". In: *ICML*. 2020.
- [57] T. Lesort. "Continual Feature Selection: Spurious Features in Continual Learning". In: *arXiv:2203.01012* (2022).
- [58] M. Abdelsalam, M. Faramarzi, et al. "IIIRC: Incremental Implicitly-Refined Classification". In: *CVPR* (2021).
- [59] N. Dzemidovich and A. Gepperth. "An empirical comparison of generators in replay-based continual learning". In: *ESANN*. 2022.
- [60] B. Pfülb, B. Bagus, et al. "Continual Learning with Fully Probabilistic Models". In: *CVPR Workshop CLVISION Findings paper*. 2021.
- [61] B. Pfülb and A. Gepperth. "Overcoming Catastrophic Forgetting with Gaussian Mixture Replay". In: *IJCNN*. 2021.
- [62] R. C. Pinto and P. M. Engel. "A fast incremental gaussian mixture model". In: *PLoS One* 10 (2015).

- [63] K. Shmelkov, C. Schmid, et al. "Incremental Learning of Object Detectors without Catastrophic Forgetting". In: 2017. arXiv:1708.06977.
- [64] D. Pokrajac, A. Lazarevic, et al. "Incremental local outlier detection for data streams". In: *Symp. on Comp. Intelligence and Data Mining*. IEEE, 2007.
- [65] M. Kristan, D. Skocaj, et al. "Incremental learning with Gaussian mixture models". In: *Computer vision winter workshop*. Slovenian Pattern Rec. Soc, 2008.
- [66] A. Gepperth and B. Pfübl. "A Rigorous Link Between Self-Organizing Maps and Gaussian Mixture Models". In: *ICANN*. 2020.
- [67] A. Gepperth. "Incremental learning with a homeostatic self-organizing neural architecture". In: *Neural Computing and Applications* (2019).
- [68] S. Vijayakumar and S. Schaal. "Locally weighted projection regression: An O(n) algorithm for incremental real time learning in high-dimensional spaces". In: *ICML*. 2000.
- [69] D. T. Pham, S. S. Dimov, et al. "An incremental K-means algorithm". In: *Journal of Mechanical Engineering Science* 7 (2004).
- [70] B. Aaron, D. E. Tamir, et al. "Dynamic incremental k-means clustering". In: *Int. Conf. on Computational Science and Comp. Intelligence*. IEEE, 2014.
- [71] A. M. Bagirov, J. Ugon, et al. "Fast modified global k-means algorithm for incremental cluster construction". In: *Pattern recognition* 4 (2011).
- [72] C. G. Turhan and H. S. Bilge. "Recent trends in deep generative models: a review". In: *Int. Conf. on Computer Science and Engineering*. IEEE, 2018.
- [73] A. Seff, A. Beatson, et al. "Continual Learning in Generative Adversarial Nets". In: arXiv:1705.08395 (2017).
- [74] M. Zhai, L. Chen, et al. "Lifelong GAN: Continual learning for conditional image generation". In: *CVPR*. 2019.
- [75] F. Wiewel and B. Yang. "Continual Learning for Anomaly Detection with Variational Autoencoder". In: *Int. Conf. on Acoustics, Speech and Signal Processing*. 2019.
- [76] D. Campo, G. Slavic, et al. "Continual Learning of Predictive Models In Video Sequences Via Variational Autoencoders". In: *Int. Conf. on Image Processing (ICIP)*. 2020.
- [77] T. Lesort, A. Gepperth, et al. "Marginal replay vs conditional replay for continual learning". In: *IJCNN*. Springer, 2019.
- [78] J. Ramapuram, M. Gregorova, et al. "Lifelong Generative Modeling". In: arXiv:1705.09847 (2017).
- [79] S. Varshney, V. K. Verma, et al. "CAM-GAN: Continual Adaptation Modules for Generative Adversarial Networks". In: *NeurIPS*. 2021.
- [80] E. Fini, V. G. T. da Costa, et al. "Self-supervised models are continual learners". In: *CVPR*. 2022.
- [81] O. Ostapenko, T. Lesort, et al. *Foundational Models for Continual Learning: An Empirical Study of Latent Replay*. 2022.
- [82] L. Caccia and J. Pineau. "SPeCiaL: Self-Supervised Pretraining for Continual Learning". In: *IJCAI, Workshop on Continual Semi-Supervised Learning*. 2021.
- [83] M. B. Ring. "Continual learning in reinforcement environments". PhD thesis. University of Texas at Austin Austin, Texas 78712, 1994.
- [84] S. Thrun and T. M. Mitchell. "Lifelong robot learning". In: *Robotics and autonomous systems* 1-2 (1995).
- [85] J. Xu and Z. Zhu. "Reinforced Continual Learning". In: 2018.
- [86] S. Zhang and R. S. Sutton. "A deeper look at experience replay". In: arXiv:1712.01275 (2017).
- [87] W. Fedus, P. Ramachandran, et al. "Revisiting fundamentals of experience replay". In: *ICML*. 2020.
- [88] T. Schaul, J. Quan, et al. "Prioritized experience replay". In: arXiv:1511.05952 (2015).
- [89] M. Andrychowicz, F. Wolski, et al. "Hindsight experience replay". In: *NeurIPS*. 2017.
- [90] D. Isele and A. Cosgun. "Selective experience replay for lifelong learning". In: *AAAI*. 2018.
- [91] G. Novati and P. Koumoutsakos. "Remember and forget for experience replay". In: *ICML*. 2019.
- [92] H. Hu, J. Ye, et al. "Generalizable Episodic Memory for Deep Reinforcement Learning". In: arXiv:2103.06469 (2021).
- [93] H. B. Ammar, E. Eaton, et al. "Online multi-task learning for policy gradient methods". In: *ICML*. 2014.
- [94] Y. Teh, V. Bapst, et al. "Distral: Robust multi-task reinforcement learning". In: *Advances in Neural Information Processing Systems* (2017).
- [95] A. Y. Sorokin and M. S. Burtsev. "Continual and multi-task reinforcement learning with shared episodic memory". In: arXiv:1905.02662 (2019).
- [96] J. Ribeiro, F. S. Melo, et al. "Multi-task Learning and Catastrophic Forgetting in Continual Reinforcement Learning". In: arXiv:1909.10008 (2019).
- [97] R. Schiewer and L. Wiskott. "Modular Networks Prevent Catastrophic Interference in Model-Based Multi-Task Reinforcement Learning". In: *Int. Conf. on Machine Learning, Optimization, and Data Science*. Springer, 2021.
- [98] R. Zhao, X. Sun, et al. "Maximum entropy-regularized multi-goal reinforcement learning". In: *ICML*. 2019.
- [99] R. Yang, H. Xu, et al. "Multi-task reinforcement learning with soft modularization". In: *NeurIPS*. 2020.
- [100] A. Gupta, J. Yu, et al. "Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention". In: *ICRA*. IEEE, 2021.
- [101] Z. Xu, K. Wu, et al. "Knowledge Transfer in Multi-Task Deep Reinforcement Learning for Continuous Control". In: *NeurIPS*. 2020.
- [102] D. Kalashnikov, J. Varley, et al. "MT-Opt: Continuous Multi-Task Robotic Reinforcement Learning at Scale". In: arXiv:2104.08212 (2021).
- [103] S. Kelly, T. Voegerl, et al. "Evolving hierarchical memory-prediction machines in multi-task reinforcement learning". In: *Genetic Programming and Evolvable Machines* 4 (2021).
- [104] S. Padakandia. "A survey of reinforcement learning algorithms for dynamically varying environments". In: *ACM Computing Surveys (CSUR)* 6 (2021).
- [105] J. A. Mendez, H. van Seijen, et al. "Modular Lifelong Reinforcement Learning via Neural Composition". In: *ICLR*. 2021.
- [106] T. Zhang, X. Wang, et al. "Catastrophic Interference in Reinforcement Learning: A Solution Based on Context Division and Knowledge Distillation". In: *IEEE Trans. on Neural Networks and Learning Systems* (2022).
- [107] C. Kaplanis, M. Shanahan, et al. "Continual reinforcement learning with complex synapses". In: *ICML*. 2018.
- [108] C. Kaplanis, M. Shanahan, et al. "Policy Consolidation for Continual Reinforcement Learning". In: arXiv:1902.00255 (2019).
- [109] Y. L. Lo and S. Ghassian. "Overcoming Catastrophic Interference in Online Reinforcement Learning with Dynamic Self-Organizing Maps". In: arXiv:1910.13213 (2019).
- [110] Y. Huang, K. Xie, et al. "Continual model-based reinforcement learning with hypernetworks". In: *ICRA*. IEEE, 2021.
- [111] A. Xie, J. Harrison, et al. "Deep Reinforcement Learning amidst Lifelong Non-Stationarity". In: arXiv:2006.10701 (2020).
- [112] J. Mendez, B. Wang, et al. "Lifelong policy gradient learning of factored policies for faster training without forgetting". In: *NeurIPS*. 2020.
- [113] C. Atkinson, B. McCane, et al. "Pseudorehearsal: Achieving deep reinforcement learning without catastrophic forgetting". In: *Neurocomputing* (2021).
- [114] K. Khetarpal, S. Sodhani, et al. "Environments for Lifelong Reinforcement Learning". In: arXiv:1811.10732 (2018).
- [115] M. Wołczyk, M. Zajac, et al. "Continual world: A robotic benchmark for continual reinforcement learning". In: *Advances in Neural Information Processing Systems*. 2021.
- [116] E. C. Johnson, E. Q. Nguyen, et al. "L2Explorer: A Lifelong Reinforcement Learning Assessment Environment". In: arXiv:2203.07454 (2022).