A Comprehensive Survey of Continual Learning: Theory, Method and Application

Liyuan Wang, Xingxing Zhang, Hang Su, *Member, IEEE*, and Jun Zhu, *Fellow, IEEE*

(Survey Paper)

***Abstract*—To cope with real-world dynamics, an intelligent system needs to incrementally acquire, update, accumulate, and exploit knowledge throughout its lifetime. This ability, known as continual learning, provides a foundation for AI systems to develop themselves adaptively. In a general sense, continual learning is explicitly limited by catastrophic forgetting, where learning a new task usually results in a dramatic performance drop of the old tasks. Beyond this, increasingly numerous advances have emerged in recent years that largely extend the understanding and application of continual learning. The growing and widespread interest in this direction demonstrates its realistic significance as well as complexity. In this work, we present a comprehensive survey of continual learning, seeking to bridge the basic settings, theoretical foundations, representative methods, and practical applications. Based on existing theoretical and empirical results, we summarize the general objectives of continual learning as ensuring a proper stability-plasticity trade-off and an adequate intra/inter-task generalizability in the context of resource efficiency. Then we provide a state-of-the-art and elaborated taxonomy, extensively analyzing how representative strategies address continual learning, and how they are adapted to particular challenges in various applications. Through an in-depth discussion of promising directions, we believe that such a holistic perspective can greatly facilitate subsequent exploration in this field and beyond.**

***Index Terms*—Continual learning, incremental learning, lifelong learning, catastrophic forgetting.**

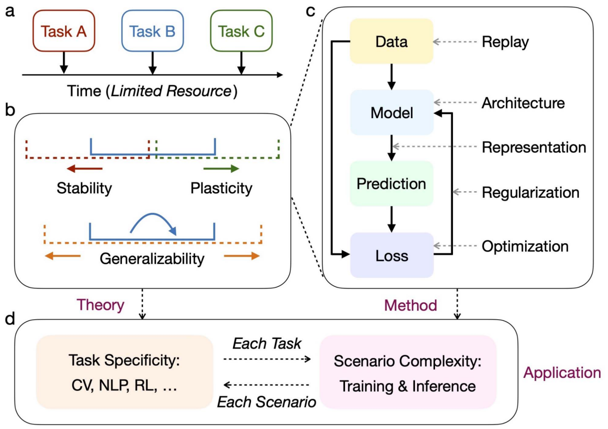


Fig. 1. Conceptual framework of continual learning. **(a)**, Continual learning requires adapting to incremental tasks with dynamic data distributions (Section II). **(b)**, A desirable solution should ensure an appropriate trade-off between stability (red arrow) and plasticity (green arrow), as well as an adequate generalizability to intra-task (blue arrow) and inter-task (orange arrow) distribution differences (Section III). **(c)**, To achieve the objective of continual learning, representative methods have targeted various aspects of machine learning (Section IV). **(d)**, Continual learning is adapted to practical applications to address particular challenges such as scenario complexity and task specificity (Section V).

I. Introduction

Learning is the basis for intelligent systems to accommodate dynamic environments. In response to external changes, evolution has empowered human and other organisms with strong adaptability to continually acquire, update, accumulate and exploit knowledge [1], [2], [3]. Naturally, we expect artificial intelligence (AI) systems to adapt in a similar way. This motivates the study of *continual learning*, where a typical setting is to learn a sequence of contents one by one and behave as if they were observed simultaneously (Fig. 1(a)). Such contents could be new skills, new examples of old skills, different environments, different contexts, etc., with particular realistic challenges incorporated [2], [4]. As the contents are provided incrementally over a lifetime, continual learning is also referred to as *incremental learning* or *lifelong learning* in much of the literature, without a strict distinction [1], [5].

Unlike conventional machine learning models built on the premise of capturing a static data distribution, continual learning is characterized by learning from dynamic data distributions. A major challenge is known as *catastrophic forgetting* [6], [7], where adaptation to a new distribution generally results in a largely reduced ability to capture the old ones. This dilemma is a facet of the trade-off between *learning plasticity* and *memory stability*: an excess of the former interferes with the latter,

Manuscript received 13 February 2023; revised 13 September 2023; accepted 5 February 2024. Date of publication 26 February 2024; date of current version 2 July 2024. This work was supported in part by the NSFC Projects under Grant 62350080, Grant 62076147, Grant 62106123, and Grant U19A2081, in part by BNRist under Grant BNR2022RC01006, in part by Tsinghua Institute for Guo Qiang, Tsinghua-OPPO Joint Research Center for Future Terminal Technology, and in part by the High Performance Computing Center, Tsinghua University. The work of Liyuan Wang was also supported in part by the Postdoctoral Fellowship Program of CPSF under Grant GZB20230350 and in part by the Shuimu Tsinghua Scholar. The work of Jun Zhu was also supported in part by the New Cornerstone Science Foundation through the XPLORER PRIZE. Recommended for acceptance by M. Kloft. *(Corresponding author: Jun Zhu.)*

The authors are with the Department of Computer Science & Technology, Institute for AI, BNRist Center, THBI Lab, Tsinghua-Bosch Joint Center for ML, Tsinghua University, Beijing 100190, China (e-mail: wly19@tsinghua.org.cn; xxzhang1993@gmail.com; suhangss@tsinghua.edu.cn; dcszj@tsinghua.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3367329, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3367329

and vice versa. Beyond simply balancing the "proportions" of these two aspects, a desirable solution for continual learning should obtain strong *generalizability* to accommodate distribution differences within and between tasks (Fig. 1(b)). As a naive baseline, reusing all old training samples (if allowed) makes it easy to address the above challenges, but creates huge computational and storage overheads, as well as potential privacy issues. In fact, continual learning is primarily intended to ensure the *resource efficiency* of model updates, preferably close to learning only new training samples.

A number of continual learning methods have been proposed in recent years for various aspects of machine learning (Fig. 1(c))., which can be conceptually grouped into five major categories: adding regularization terms with reference to the old model (regularization-based approach); approximating and recovering the old data distributions (replay-based approach); explicitly manipulating the optimization programs (optimization-based approach); learning robust and well-generalized representations (representation-based approach); and constructing task-specific parameters with a properly-designed architecture (architecture-based approach). This taxonomy extends the commonly-used ones with current advances, and provides refined sub-directions for each category. We summarize how these methods achieve the objective of continual learning, with an extensive analysis of their theoretical foundations and specific implementations. In particular, these methods are *closely connected*, e.g., regularization and replay ultimately act to rectify the gradient directions in optimization, and *highly synergistic*, e.g., the efficacy of replay can be facilitated by distilling knowledge from the old model.

Realistic applications present particular challenges for continual learning, categorized into *scenario complexity* and *task specificity* (Fig. 1(d)). As for the former, for example, the task identity is probably missing in training and testing, and the training samples might be introduced in tiny batches or even one pass. Due to the expense and scarcity of data labeling, continual learning needs to be effective for few-shot, semi-supervised and even unsupervised scenarios. As for the latter, although current advances mainly focus on visual classification, other representative visual domains such as object detection and semantic segmentation, as well as other related fields such as conditional generation, reinforcement learning (RL) and natural language processing (NLP), are receiving increasing attention with their own characteristics. We summarize these particular challenges and analyze how continual learning methods are adapted to them.

Considering the significant growth of interest in continual learning, we believe that such an *up-to-date* and *comprehensive* survey can provide a holistic perspective for subsequent work. Although there are some early surveys of continual learning with relatively broad coverage [2], [5], [8], important advances in recent years have not been incorporated. In contrast, the latest surveys typically collate only a local aspect of continual learning, with respect to its biological underpinnings [1], [3], [9], [10], specialized settings for visual classification [11], [12], [13], [14], and specific extensions for NLP [15], [16] or RL [17]. We include a detailed comparison of previous surveys in Appendix Table 1, available online. These surveys are typically hard to be both up-to-date and comprehensive, which is the primary strength of this work.

Compared to previous surveys, our improvements lie in the following aspects, including (1) Setup: collect and formulate more typical scenarios that have emerged in recent years; (2) Theory: summarize theoretical efforts on continual learning in terms of stability, plasticity and generalizability; (3) Method: add optimization-based and representation-based approaches on the top of regularization-based, replay-based and architecture-based approaches, with an extensive analysis of their sub-directions; (4) Application: summarize practical applications of continual learning and their particular challenges in terms of scenario complexity and task specificity; (5) Linkage: discuss underlying connections between theory, method and application, as well as promising crossovers with other related fields.

The organization of this paper is described in Fig. 1. We introduce basic setups of continual learning in Section II, summarize theoretical efforts for its general objectives in Section III, present a state-of-the-art and elaborated taxonomy of representative methods in Section IV, describe how these methods are adapted to practical challenges in Section V, and discuss promising directions for subsequent work in Section VI.

II. Setup

In this section, we first present a basic formulation of continual learning. Then we introduce typical scenarios and evaluation metrics.

A. Basic Formulation

Continual learning is characterized as learning from dynamic data distributions. In practice, training samples of different distributions arrive in sequence. A continual learning model parameterized by θ needs to learn corresponding task(s) with no or limited access to old training samples and perform well on their test sets. Formally, an incoming batch of training samples belonging to a task t can be represented as $\mathcal{D}_{t,b} = \{\mathcal{X}_{t,b}, \mathcal{Y}_{t,b}\}$, where $\mathcal{X}_{t,b}$ is the input data, $\mathcal{Y}_{t,b}$ is the data label, $t \in \mathcal{T} = \{1, \ldots, k\}$ is the task identity and $b \in \mathcal{B}_t$ is the batch index ($\mathcal{T}$ and $\mathcal{B}_t$ denote their space, respectively). Here we define a "task" by its training samples $\mathcal{D}_t$ following the distribution $\mathbb{D}_t := p(\mathcal{X}_t, \mathcal{Y}_t)$ ($\mathcal{D}_t$ denotes the entire training set by omitting the batch index, likewise for $\mathcal{X}_t$ and $\mathcal{Y}_t$), and assume that there is no difference in distribution between training and testing. Under realistic constraints, the data label $\mathcal{Y}_t$ and the task identity t might not be always available. In continual learning, the training samples of each task can arrive incrementally in batches (i.e., $\{\{\mathcal{D}_{t,b}\}_{b \in \mathcal{B}_t}\}_{t \in \mathcal{T}}$) or simultaneously (i.e., $\{\mathcal{D}_t\}_{t \in \mathcal{T}}$).

B. Typical Scenario

According to the division of incremental batches and the availability of task identities, we detail the typical scenarios as follows (see Table I for a formal comparison):

- *Instance-Incremental Learning* (IIL)*:* All training samples belong to the same task and arrive in batches.
- *Domain-Incremental Learning* (DIL)*:* Tasks have the same data label space but different input distributions. Task identities are not required.

TABLE I
A FORMAL COMPARISON OF TYPICAL CONTINUAL LEARNING SCENARIOS

Scenario	Training	Testing
IIL [18]	$\{\{\mathcal{D}_{t,b}, t\}_{b\in\mathcal{B}_t}\}_{t=j}$	$\{p(\mathcal{X}_t)\}_{t=j}$; t is not required
DIL [4], [19]	$\{\mathcal{D}_t, t\}_{t\in\mathcal{T}}$; $p(\mathcal{X}_i) \neq p(\mathcal{X}_j)$ and $\mathcal{Y}_i = \mathcal{Y}_j$ for $i \neq j$	$\{p(\mathcal{X}_t)\}_{t\in\mathcal{T}}$, t is not required
TIL [4], [19]	$\{\mathcal{D}_t, t\}_{t\in\mathcal{T}}$; $p(\mathcal{X}_i) \neq p(\mathcal{X}_j)$ and $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$	$\{p(\mathcal{X}_t)\}_{t\in\mathcal{T}}$; t is available
CIL [4], [19]	$\{\mathcal{D}_t, t\}_{t\in\mathcal{T}}$; $p(\mathcal{X}_i) \neq p(\mathcal{X}_j)$ and $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$	$\{p(\mathcal{X}_t)\}_{t\in\mathcal{T}}$; t is unavailable
TFCL [20]	$\{\{\mathcal{D}_{t,b}\}_{b\in\mathcal{B}_t}\}_{t\in\mathcal{T}}$; $p(\mathcal{X}_i) \neq p(\mathcal{X}_j)$ and $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$	$\{p(\mathcal{X}_t)\}_{t\in\mathcal{T}}$; t is optionally available
OCL [21]	$\{\{\mathcal{D}_{t,b}\}_{b\in\mathcal{B}_t}\}_{t\in\mathcal{T}}$, $\lvert b\rvert = 1$; $p(\mathcal{X}_i) \neq p(\mathcal{X}_j)$ and $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$	$\{p(\mathcal{X}_t)\}_{t\in\mathcal{T}}$; t is optionally available
CPT [22]	$\{\mathcal{D}_t^{pt}, t\}_{t\in\mathcal{T}^{pt}}$, followed by a downstream task j	$\{p(\mathcal{X}_t)\}_{t=j}$; t is not required

$\mathcal{D}_{t,b}$: the training samples of task t and batch b. $|b|$: the size of batch b. $\mathcal{B}_t$: the space of incremental batches belonging to task t. $\mathcal{D}_t$: the training set of task t (further specified as $\mathcal{D}_t^{pt}$ for pre-training). $\mathcal{T}$: the space of all incremental tasks (further specified as $\mathcal{T}^{pt}$ for pre-training). $\mathcal{X}_t$: the input data in $\mathcal{D}_t$. $p(\mathcal{X}_t)$: the distribution of $\mathcal{X}_t$. $\mathcal{Y}_t$: the data label of $\mathcal{X}_t$.

- *Task-Incremental Learning* (TIL)*:* Tasks have disjoint data label spaces. Task identities are provided in both training and testing.
- *Class-Incremental Learning* (CIL)*:* Tasks have disjoint data label spaces. Task identities are only provided in training.
- *Task-Free Continual Learning* (TFCL)*:* Tasks have disjoint data label spaces. Task identities are not provided in either training or testing.
- *Online Continual Learning* (OCL)*:* Tasks have disjoint data label spaces. Training samples for different tasks arrive as a one-pass data stream.
- *Continual Pre-training* (CPT)*:* Pre-training data arrives in sequence. The goal is to improve knowledge transfer to downstream tasks.

If not specified, each task is often assumed to have a sufficient number of labeled training samples, i.e., *Supervised Continual Learning*. According to the provided $\mathcal{X}_t$ and $\mathcal{Y}_t$ in each $\mathcal{D}_t$, continual learning is further extended to zero-shot [23], few-shot [24], semi-supervised [25], open-world [26] and un-/self-supervised [27], [28] scenarios. Besides, other realistic considerations have been incorporated, such as multiple labels [29], noisy labels [30], blurred boundary [31], [32], hierarchical granularity [33] and sub-populations [34], mixture of task similarity [35], long-tailed distribution [36], novel class discovery [37], [38], multi-modality [39], etc. Some recent work has focused on combinatorial scenarios [18], [40], [41], [42], [43] in order to simulate real-world complexity.

C. Evaluation Metric

In general, the performance of continual learning can be evaluated from three aspects: overall performance of the tasks learned so far, memory stability of old tasks, and learning plasticity of new tasks. Here we summarize the common evaluation metrics, using classification as an example.

Overall performance is typically evaluated by *average accuracy* (AA) [44], [45] and *average incremental accuracy* (AIA) [46], [47]. Let $a_{k,j} \in [0, 1]$ denote the classification accuracy evaluated on the test set of the jth task after incremental learning of the kth task ($j \leq k$). The output space to compute $a_{k,j}$ consists of the classes in either $\mathcal{Y}_j$ or $\cup_{i=1}^k \mathcal{Y}_i$, corresponding to the use of multi-head evaluation (e.g., TIL) or single-head evaluation (e.g., CIL) [44]. The two metrics at the kth task are then defined as $\mathrm{AA}_k = \frac{1}{k}\sum_{j=1}^k a_{k,j}$ and $\mathrm{AIA}_k = \frac{1}{k}\sum_{i=1}^k \mathrm{AA}_i$, where AA represents the overall performance at the current moment and AIA further reflects the historical performance.

Memory stability can be evaluated by *forgetting measure* (FM) [44] and *backward transfer* (BWT) [45]. As for the former, the forgetting of a task is calculated by the difference between its maximum performance obtained in the past and its current performance: $f_{j,k} = \max_{i\in\{1,\ldots,k-1\}}(a_{i,j} - a_{k,j}), \forall j < k$. FM at the kth task is the average forgetting of all old tasks: $\mathrm{FM}_k = \frac{1}{k-1}\sum_{j=1}^{k-1} f_{j,k}$. As for the latter, BWT evaluates the average influence of learning the kth task on all old tasks: $\mathrm{BWT}_k = \frac{1}{k-1}\sum_{j=1}^{k-1}(a_{k,j} - a_{j,j})$, where the forgetting is generally reflected as a negative BWT.

Learning plasticity can be evaluated by *intransience measure* (IM) [44] and *forward transfer* (FWT) [45]. IM is defined as the inability of a model to learn new tasks, which is calculated by the difference of a task between its joint training performance and continual learning performance: $\mathrm{IM}_k = a_k^* - a_{k,k}$, where a_k^* is the classification accuracy of a randomly-initialized reference model jointly trained with $\cup_{j=1}^k \mathcal{D}_j$ for the kth task. In comparison, FWT evaluates the average influence of all old tasks on the current kth task: $\mathrm{FWT}_k = \frac{1}{k-1}\sum_{j=2}^k (a_{j,j} - \tilde{a}_j)$, where $\tilde{a}_j$ is the classification accuracy of a randomly-initialized reference model trained with $\mathcal{D}_j$ for the jth task. Note that, $a_{k,j}$ can be adapted to other forms depending on the task type, and the above evaluation metrics should be adjusted accordingly.

III. THEORY

In this section, we summarize the theoretical efforts on continual learning, with respect to both stability-plasticity trade-off and generalizability analysis, and relate them to the motivations of various continual learning methods.

A. Stability-Plasticity Trade-Off

With the basic formulation in Section II-A, let's consider a general setup for continual learning, where a neural

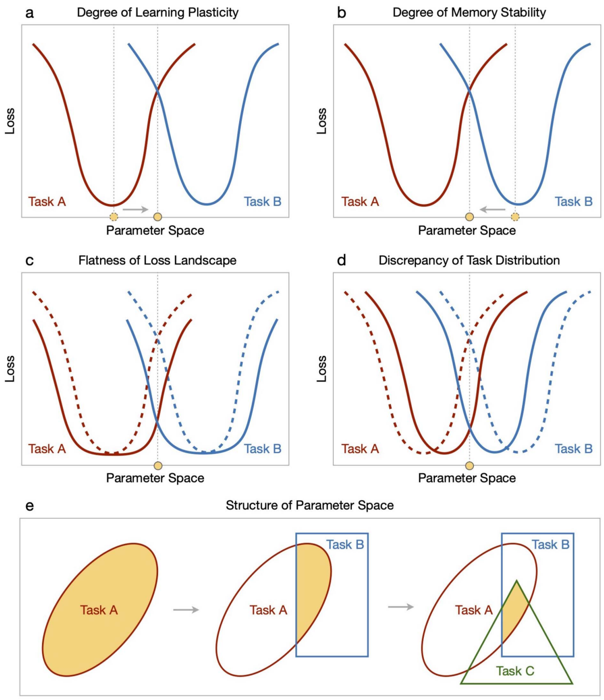


Fig. 2. Analysis of critical factors for continual learning. **(a)**, **(b)**, Continual learning requires a proper balance between learning plasticity and memory stability, where excess of either can affect the overall performance. **(c)**, **(d)**, When the converged loss landscape is flatter and the observed data distributions are more similar, a properly balanced solution can better generalize to the task sequence. **(e)**, The structure of parameter space determines the complexity and possibility of finding a desirable solution (adapted from [62]). The yellow area indicates the feasible parameter space shared by individual tasks, which tends to be narrow and irregular as more incremental tasks are introduced.

network with parameters $\theta \in \mathbb{R}^{|\theta|}$ needs to learn k incremental tasks. The training set and test set of each task are assumed to follow the same distribution $\mathbb{D}_t$, $t = 1, \ldots, k$, where the training set $\mathcal{D}_t = \{\mathcal{X}_t, \mathcal{Y}_t\} = \{(x_{t,n}, y_{t,n})\}_{n=1}^{N_t}$ includes N_t data-label pairs. The objective is to learn a probabilistic model $p(\mathcal{D}_{1:k}|\theta) = \prod_{t=1}^{k} p(\mathcal{D}_t|\theta)$ that can perform well on all tasks denoted as $\mathcal{D}_{1:k} := \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$. The task-related performance for discriminative models can be expressed as $\log p(\mathcal{D}_t|\theta) = \sum_{n=1}^{N_t} \log p_\theta(y_{t,n}|x_{t,n})$. The central challenge of continual learning generally arises from the sequential nature of learning: when learning the kth task from $\mathcal{D}_k$, the old training sets $\{\mathcal{D}_1, \ldots, \mathcal{D}_{k-1}\}$ are inaccessible. Therefore, it is critical but extremely challenging to capture the distributions of both old and new tasks in a balanced manner, i.e., ensuring a proper *stability-plasticity trade-off*, where excessive learning plasticity or memory stability can largely compromise each other (Fig. 2(a) and (b)).

A straightforward idea is to approximate and recover the old data distributions by storing a few old training samples or training a generative model, known as the *replay-based approach* (Section IV-B). According to the learning theory for supervised learning [48], the performance of an old task is improved with replaying more old training samples that approximate its distribution, but resulting in potential privacy issues and a linear increase in *resource overhead*. The use of generative models is also limited by a huge additional resource overhead, as well as their own catastrophic forgetting and expressiveness.

An alternative choice is to propagate the old data distributions in updating parameters through formulating continual learning in a Bayesian framework. Based on a prior $p(\theta)$ of the network parameters, the posterior after observing the kth task can be computed with Bayes' rule

$$p(\theta|\mathcal{D}_{1:k}) \propto p(\theta) \prod_{t=1}^{k} p(\mathcal{D}_t|\theta) \propto p(\theta|\mathcal{D}_{1:k-1}) p(\mathcal{D}_k|\theta), \quad (1)$$

where the posterior $p(\theta|\mathcal{D}_{1:k-1})$ of the (k-1)th task becomes the prior of the kth task, and thus enables the new posterior $p(\theta|\mathcal{D}_{1:k})$ to be computed with only the current training set $\mathcal{D}_k$. However, as the posterior is generally intractable (except very special cases), a common option is to approximate it with $q_{k-1}(\theta) \approx p(\theta|\mathcal{D}_{1:k-1})$, likewise for $q_k(\theta) \approx p(\theta|\mathcal{D}_{1:k})$. In the following, we will introduce two widely-used approximation strategies:

The first is online *Laplace approximation*, which approximates $p(\theta|\mathcal{D}_{1:k-1})$ as a multivariate Gaussian with local gradient information [49], [50], [51], [52]. Specifically, we can parameterize $q_{k-1}(\theta)$ with ϕ_{k-1} and construct an approximate Gaussian posterior $q_{k-1}(\theta) := q(\theta; \phi_{k-1}) = \mathcal{N}(\theta; \mu_{k-1}, \Lambda_{k-1}^{-1})$ through performing a second-order Taylor expansion around the mode $\mu_{k-1} \in \mathbb{R}^{|\theta|}$ of $p(\theta|\mathcal{D}_{1:k-1})$, where Λ_{k-1} denotes the precision matrix and $\phi_{k-1} = \{\mu_{k-1}, \Lambda_{k-1}\}$, likewise for $q(\theta; \phi_k)$, μ_k and Λ_k. According to (1), the posterior mode for learning the current kth task can be computed as

$$\begin{aligned} \mu_k &= \arg\max\nolimits_\theta \log p(\theta|\mathcal{D}_{1:k}) \\ &\approx \arg\max\nolimits_\theta \log p(\mathcal{D}_k|\theta) - \frac{1}{2}(\theta - \mu_{k-1})^\top \Lambda_{k-1} (\theta - \mu_{k-1}), \end{aligned} \quad (2)$$

which is updated recursively from μ_{k-1} and Λ_{k-1}. Meanwhile, Λ_k is updated recursively from Λ_{k-1}

$$\begin{aligned} \Lambda_k &= -\nabla_\theta^2 \log p(\theta|\mathcal{D}_{1:k})\big|_{\theta=\mu_k} \\ &\approx -\nabla_\theta^2 \log p(\mathcal{D}_k|\theta)\big|_{\theta=\mu_k} + \Lambda_{k-1}, \end{aligned} \quad (3)$$

where the first term on the right side is the Hessian of the negative log likelihood of $\mathcal{D}_k$ at μ_k, denoted as $H(\mathcal{D}_k, \mu_k)$. In practice, $H(\mathcal{D}_k, \mu_k)$ is often computationally inefficient due to the great dimensionality of $\mathbb{R}^{|\theta|}$, and there is no guarantee that the approximated Λ_k is positive semi-definite for the Gaussian assumption. To overcome these issues, the Hessian can be approximated by the Fisher information matrix (FIM)

$$F_k = \mathbb{E}[\nabla_\theta \log p(\mathcal{D}_k|\theta) \nabla_\theta \log p(\mathcal{D}_k|\theta)^\top]\big|_{\theta=\mu_k} \approx H(\mathcal{D}_k, \mu_k). \quad (4)$$

For ease of computation, the FIM can be further simplified with a diagonal approximation [49], [53] or a Kronecker-factored approximation [50], [54]. Then, (2) is implemented by saving a frozen copy of the old model μ_{k-1} to regularize parameter changes, known as the *regularization-based approach* (Section IV-A). Here, we use EWC [49] as an example and present its loss function

$$\mathcal{L}_{\text{EWC}}(\theta) = \ell_k(\theta) + \frac{\lambda}{2}(\theta - \mu_{k-1})^\top \hat{F}_{1:k-1}(\theta - \mu_{k-1}), \quad (5)$$

where ℓ_k denotes the task-specific loss, the FIM $\hat{F}_{1:k-1} = \sum_{t=1}^{k-1} \text{diag}(F_t)$ with a diagonal approximation $\text{diag}(\cdot)$ of each F_t, and λ is a hyperparameter to control the strength.

The second is online *variational inference* (VI) [55], [56], [57], [58], [59], [60]. A representative way for online VI is to minimize the following KL-divergence over a family $\mathcal{Q}$ that satisfies $p(\theta|\mathcal{D}_{1:k}) \in \mathcal{Q}$ at the current kth task:

$$q_k(\theta) = \arg\min_{q \in \mathcal{Q}} \text{KL}(q(\theta) \parallel \frac{1}{Z_k} q_{k-1}(\theta) p(\mathcal{D}_k|\theta)), \quad (6)$$

where Z_k is the normalizing constant of $q_{k-1}(\theta)p(\mathcal{D}_k|\theta)$. In practice, the above minimization can be achieved by using an additional Monte Carlo approximation, with specifying $q_k(\theta) := q(\theta; \phi_k) = \mathcal{N}(\theta; \mu_k, \Lambda_k^{-1})$ as a multivariate Gaussian. Here we use VCL [55] as an example, which minimizes the following objective (i.e., maximize its negative):

$$\mathcal{L}_{\text{VCL}}(q_k(\theta)) = \mathbb{E}_{q_k(\theta)}(\ell_k(\theta)) + \text{KL}(q_k(\theta) \parallel q_{k-1}(\theta)), \quad (7)$$

where the KL-divergence can be computed in a closed-form and serves as an implicit regularization term. In particular, although the loss functions of (5) and (7) take similar forms, the former is a local approximation at a set of deterministic parameters θ, while the latter is computed by sampling from the variational distribution $q_k(\theta)$. This is attributed to the fundamental difference between the two approximation strategies [55], [61].

In essence, the constraint on continual learning for either replay or regularization is ultimately reflected in gradient directions. As a result, some recent work directly manipulates the gradient-based optimization process, categorized as the *optimization-based approach* (Section IV-C). Specifically, when a few old training samples $\mathcal{M}_t$ for task t are maintained in a memory buffer, gradient directions of the new training samples are encouraged to stay close to that of the $\mathcal{M}_t$ [45], [63], [64]. This is formulated as $\langle \nabla_\theta \mathcal{L}_k(\theta; \mathcal{D}_k), \nabla_\theta \mathcal{L}_k(\theta; \mathcal{M}_t) \rangle \geq 0$ for $t \in \{1, \ldots, k-1\}$, so as to essentially enforce non-increase in the loss of old tasks, i.e., $\mathcal{L}_k(\theta; \mathcal{M}_t) \leq \mathcal{L}_k(\theta_{k-1}; \mathcal{M}_t)$, where θ_{k-1} is the network parameters at the end of learning the (k-1)th task.

Alternatively, gradient projection can also be performed without storing old training samples [51], [65], [66], [67]. Here we take NCL [51] as an example, which can manipulate gradient directions with μ_{k-1} and Λ_{k-1} in online Laplace approximation. As shown in (8), NCL performs continual learning by minimizing the task-specific loss $\ell_k(\theta)$ within a region of radius r centered around θ with a distance metric $d(\theta, \theta + \delta) = \sqrt{\delta^\top \Lambda_{k-1} \delta / 2}$ that takes into account the curvature of the prior via its precision matrix Λ_{k-1}

$$\delta^* = \arg\min_{\delta} \ell_k(\theta + \delta) \approx \arg\min_{\delta} \ell_k(\theta) + \nabla_\theta \ell_k(\theta)^\top \delta,$$
$$\text{s.t.}, d(\theta, \theta + \delta) = \sqrt{\delta^\top \Lambda_{k-1} \delta / 2} \leq r. \quad (8)$$

The solution to such an optimization problem in (8) is given by $\delta^* \propto \Lambda_{k-1}^{-1} \nabla_\theta \ell_k(\theta) - (\theta - \mu_{k-1})$, which gives rise to the following update rule for a learning rate λ:

$$\theta \leftarrow \theta + \lambda[\Lambda_{k-1}^{-1} \nabla_\theta \ell_k(\theta) - (\theta - \mu_{k-1})], \quad (9)$$

where the first term encourages the parameter changes predominantly in directions that do not interfere with the old tasks via a preconditioner Λ_{k-1}^{-1}, while the second term enforces θ to stay close to the old task solution μ_{k-1}.

Of note, the above analyses are mainly based on finding a shared solution for all tasks, which is subject to severe inter-task interference [52], [68], [69]. In contrast, incremental tasks can also be learned in a (partially) separated way, which is the dominant idea of the *architecture-based approach* (Section IV-E). This can be formulated as constructing a continual learning model with parameters $\theta = \cup_{t=1}^{k} \theta^{(t)}$, where $\theta^{(t)} = \{e^{(t)}, \psi\}$, $e^{(t)}$ is the task-specific parameters, and ψ is the task-sharing parameters. The task-sharing parameters ψ are omitted in some cases, where the task-specific parameters $e^{(i)}$ and $e^{(j)}$ $(i < j)$ may overlap to enable parameter reuse and knowledge transfer. The overlapping part $e^{(i)} \cap e^{(j)}$ is usually frozen when learning the jth task to avoid catastrophic forgetting [70], [71]. Then, each task can be performed as $p(\mathcal{D}_t|\theta^{(t)})$ instead of $p(\mathcal{D}_t|\theta)$ if given the task identity $\mathbb{I}_{\mathcal{D}_t}$, in which the conflicts between tasks can be explicitly controlled or even completely avoided

$$\begin{aligned} p(\mathcal{D}_t|\theta) &= \sum_{i=1}^{k} p(\mathcal{D}_t|\mathbb{I}_{\mathcal{D}_t} = i, \theta) p(\mathbb{I}_{\mathcal{D}_t} = i|\theta) \\ &= p(\mathcal{D}_t|\mathbb{I}_{\mathcal{D}_t} = t, \theta) p(\mathbb{I}_{\mathcal{D}_t} = t|\mathcal{D}_t, \theta) \\ &= p(\mathcal{D}_t|\theta^{(t)}) p(\mathbb{I}_{\mathcal{D}_t} = t|\mathcal{D}_t, \theta). \end{aligned} \quad (10)$$

However, there are two major challenges. The first is the scalability of model size due to the progressive allocation of $\theta^{(t)}$, which depends on the sparsity of $e^{(t)}$, reusability of $e^{(i)} \cap e^{(j)}$ $(i < j)$, and transferability of ψ. The second is the accuracy of task-identity prediction, denoted as $p(\mathbb{I}_{\mathcal{D}_t} = t|\mathcal{D}_t, \theta)$. Except for the TIL setting that always provides the task identity $\mathbb{I}_{\mathcal{D}_t}$ [70], [71], [72], [73], other scenarios generally require the model to determine which $\theta^{(t)}$ to use based on the input data, as shown in (10). This is closely related to the out-of-distribution (OOD) detection, where the predictive uncertainty should be low for in-distribution data and high for OOD data [74], [75], [76], [77]. More importantly, since the function of task-identity prediction as (11) needs to be continually updated, it also suffers from catastrophic forgetting. To address this issue, the ith task's distribution $p(\mathcal{D}_t|i, \theta)$ could be recovered by replay [74], [78], [79], [80]

$$p(\mathbb{I}_{\mathcal{D}_t} = i|\mathcal{D}_t, \theta) \propto p(\mathcal{D}_t|i, \theta) p(i), \quad (11)$$

where the marginal task distribution $p(i) \propto N_i$ in general.

B. Generalizability Analysis

Current theoretical efforts for continual learning have primarily been performed on training sets of incremental tasks, assuming that their test sets follow similar distributions and the candidate solutions have similar generalizability. However, since the objective for learning multiple tasks is typically highly non-convex, there are many local optimal solutions that perform similarly on training sets but have significantly different generalizability on test sets [69], [81]. For continual learning, a desirable solution requires not only *intra-task generalizability* from training sets to test sets, but also *inter-task generalizability* to accommodate incremental changes of their distributions. Here we provide a conceptual demonstration with a task-specific loss $\ell_t(\theta; \mathcal{D}_t)$ and its empirical optimal solution $\theta_t^* = \arg\min_\theta \ell_t(\theta; \mathcal{D}_t)$. When task i needs to accommodate another task j, the maximum sacrifice of its performance can be estimated by performing a second-order Taylor expansion of $\ell_i(\theta; \mathcal{D}_i)$ around θ_i^*

$$\begin{aligned}\ell_i(\theta_j^*; \mathcal{D}_i) &\approx \ell_i(\theta_i^*; \mathcal{D}_i) + (\theta_j^* - \theta_i^*)^\top \nabla_\theta \ell_i(\theta; \mathcal{D}_i)\big|_{\theta=\theta_i^*} \\ &\quad + \frac{1}{2}(\theta_j^* - \theta_i^*)^\top \nabla_\theta^2 \ell_i(\theta; \mathcal{D}_i)\big|_{\theta=\theta_i^*}(\theta_j^* - \theta_i^*) \\ &\approx \ell_i(\theta_i^*; \mathcal{D}_i) + \frac{1}{2}\Delta\theta^\top \nabla_\theta^2 \ell_i(\theta; \mathcal{D}_i)\big|_{\theta=\theta_i^*}\Delta\theta, \qquad (12)\end{aligned}$$

where $\Delta\theta := \theta_j^* - \theta_i^*$ and $\nabla_\theta \ell_i(\theta; \mathcal{D}_i)\big|_{\theta=\theta_i^*} \approx \mathbf{0}$. Then, the performance degradation of task i is upper-bounded by

$$\ell_i(\theta_j^*; \mathcal{D}_i) - \ell_i(\theta_i^*; \mathcal{D}_i) \leq \frac{1}{2}\lambda_i^{\max}\|\Delta\theta\|^2, \qquad (13)$$

where $\lambda_i^{\max}$ is the maximum eigenvalue of the Hessian matrix $\nabla_\theta^2 \ell_i(\theta; \mathcal{D}_i)\big|_{\theta=\theta_i^*}$. Note that the order of task i and j can be arbitrary, that is, (13) demonstrates both forward and backward effects. Therefore, the robustness of an empirical optimal solution θ_i^* to parameter changes is closely related to $\lambda_i^{\max}$, which has been a common metric to describe the *flatness of loss landscape* [81], [82], [83].

Intuitively, convergence to a local minima with a flatter loss landscape will be less sensitive to modest parameter changes and thus benefit both old and new tasks (see Fig. 2(c)). To find such a *flat minima*, there are three widely-used strategies in continual learning. The first is derived from its definition, i.e., the flatness metric. Specifically, the minimization of $\ell_t(\theta; \mathcal{D}_t)$ can be replaced by a robust task-specific loss $\ell_t^b(\theta; \mathcal{D}_t) := \max_{\|\delta\| \leq b} \ell_t(\theta + \delta; \mathcal{D}_t)$, and thus the obtained solution guarantees low error not only at a specific point but also in its neighborhood with a "radius" of b. However, due to the great dimensionality of θ, the calculation of $\ell_t^b(\theta; \mathcal{D}_t)$ cannot cover all possible δ but only a few directions [84], similar to the complexity issue of computing the Hessian matrix in (12). The alternatives include using an approximation of the Hessian [81], [85] or calculating δ only along the trajectory of forward and backward parameter changes [86], [87]. The second is to operate the loss landscape by constructing an ensemble model under the restriction of mode connectivity, i.e., integrating multiple minima in parameter or function space along the low-error path, as connecting them ensures flatness on that path [69], [86], [88]. These two strategies are closely related to the *optimization-based approach*. The third comes down to obtaining well-distributed representations, which tend to be more robust to distribution differences in function space, such as by using large-scale pre-training [87], [89], [90] and self-supervised learning [28], [91], [92], [93]. This motivates the *representation-based approach* in Section IV-D.

There are many other factors important for continual learning performance. As shown in (13), the upper bound of performance degradation also depends on the difference of the empirical optimal solution $\theta_t^* = \arg\min_\theta \ell_t(\theta; \mathcal{D}_t)$ for each task, i.e., the *discrepancy of task distribution* (see Fig. 2(d)), which is further validated by a theoretical analysis of the forgetting-generalization trade-off [94] and the PAC-Bayes bound of generalization errors [69], [95]. Therefore, how to exploit task similarity is directly related to the performance of continual learning. The generalization error for each task can improve with synergistic tasks but deteriorate with competing tasks [68], where learning all tasks equally in a shared solution tends to compromise each task in performance [68], [69]. On the other hand, when model parameters are *not* shared by all tasks (e.g., using a multi-head output layer), the impact of task similarity on continual learning will be complex. Some theoretical studies with the neural tangent kernel (NTK) [96], [97], [98], [99] suggest that an increase in task similarity may lead to more forgetting. Since the output heads are independent for individual tasks, it becomes much more difficult to distinguish between two similar solutions [98], [99]. Specifically, under the NTK regime from the tth task up until the kth task, the forgetting of old tasks is bounded by

$$\begin{aligned}&\|p(\mathcal{D}_k|\theta_k^*) - p(\mathcal{D}_k|\theta_t^*)\|_F^2 \\ &\leq \sigma_{t,|\mathrm{rep}|+1}^2 \sum_{i=t+1}^{k} \left\|\Theta^{t\rightarrow S(i,|\mathrm{rep}|)}\right\|_2^2 \left\|\Theta^{i\rightarrow S(i,|\mathrm{rep}|)}\right\|_2^2 \|\mathrm{RES_i}\|_2^2 .\end{aligned} \qquad (14)$$

$\Theta^{t\rightarrow k}$ is a diagonal matrix where each diagonal element $\cos(\theta_{t,k})_r$ is the cosine of the rth principal angle between the tth and kth tasks in the feature space. $\sigma_{t,\cdot}$ is the $\cdot$th singular value of the tth task. RES_i is the rotated residuals that remain to be learned, and $S(i, \cdot)$ represents the residuals subspace of order $\cdot$ until the ith task. $|\mathrm{rep}|$ is the sample number of replay data. The complex impact of task similarity suggests the importance of model architectures for coordinating task-sharing and task-specific components.

Moreover, the complexity of finding a desirable solution for continual learning is determined to a large extent by the *structure of parameter space*. Learning all incremental tasks with a shared solution is equivalent to learning each new task in a constrained parameter space that prevents performance degradation of all old tasks. Such a classical continual learning problem has proven to be NP-hard in general [62], because the feasible parameter space tends to be narrow and irregular as more tasks are introduced, thus difficult to identify (see Fig. 2(e)). This challenging issue can be mitigated by replaying representative old training samples [62], restricting the feasible parameter space to a hyper-rectangle [100], or alternating the model architecture of using a single parameter space (e.g., using multiple continual learning models) [68], [69], [101]. To harmonize the important factors in continual learning, recent work presents a similar form of generalized bounds for learning and forgetting. For example,

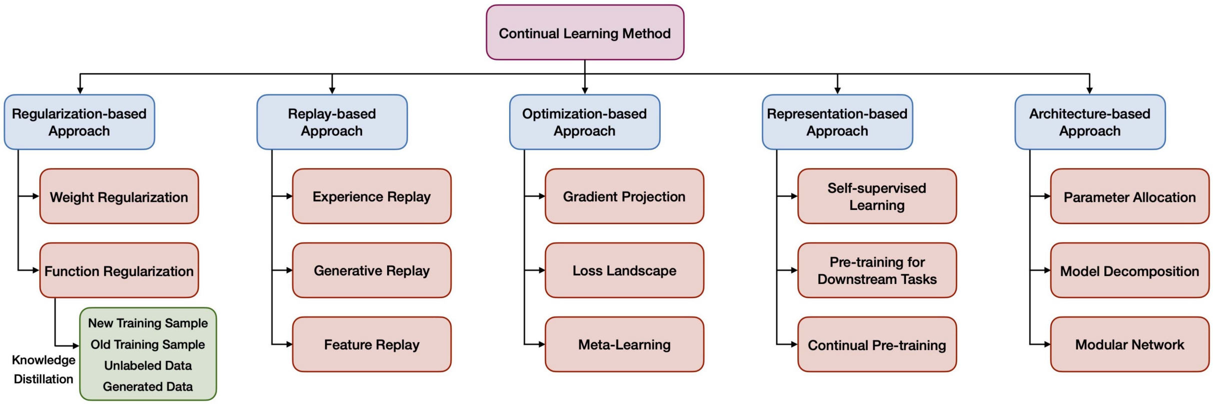


Fig. 3. State-of-the-art and elaborated taxonomy of representative continual learning methods. We have summarized five main categories (blue blocks), each of which is further divided into several sub-directions (red blocks).

with probability at least $1-\delta$ for any $\delta \in (0,1)$, an ideal continual learner [102] under the assumption that all tasks share a global minimizer with uniform convergence (i.e., $\lambda_i^{\max} = \lambda$ for $\forall t = 1,\ldots,k$ in (13)) has the generalization bound

$$c_t^* \le \mathbb{E}_{\mathcal{D}_t\sim\mathbb{D}_t}\ell_t(\theta;\mathcal{D}_t) \le c_t^* + \zeta(N_t,\delta), \forall t = 1,\ldots,k, \quad (15)$$

where $c_t^* = \ell_t(\theta_t^*;\mathcal{D}_t)$ is the minimum loss of the tth task, and θ is a global solution of the continually learned $1:k$ tasks by empirical risk minimization. $\zeta = O(\frac{\lambda B\sqrt{|\theta|\log(N_t)\log(|\theta|k/\delta)}}{2\sqrt{N_t}})$, and $\|\theta\|_2 \le B$. Considering that the shared parameter space for many different tasks might be an empty set (Fig. 2(e)), i.e., $\cup_{t=1}^k \theta_t = \emptyset$, the generalization bounds are further refined by assuming K parameter spaces ($K \ge 1$ in general) to capture all tasks [69], [103]. For generalization errors of new and old tasks

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}_t\sim\mathbb{D}_t}\ell_t(\theta;\mathcal{D}_t) \le c_t^* + R\left(\sum_{i=1}^{t-1}\ell_i^b\right)\\&+\sum_{\mathrm{i=1}}^{\mathrm{t-1}}\mathrm{Div}(\mathbb{D}_\mathrm{i},\mathbb{D}_\mathrm{t}) + \zeta\left(\sum_{\mathrm{i=1}}^{\mathrm{t-1}}\mathrm{N_i},\mathrm{K}/\delta\right),\\&\sum_{i=1}^{t-1}\mathbb{E}_{\mathcal{D}_i\sim\mathbb{D}_i}\ell_i(\theta;\mathcal{D}_i) \le \sum_{i=1}^{t-1}c_i^* + R(\ell_t^b) + \sum_{i=1}^{t-1}\mathrm{Div}(\mathbb{D}_\mathrm{t},\mathbb{D}_\mathrm{i})\\&+\zeta(N_t,K/\delta), \qquad (16)\end{aligned}$$

where $R(\cdot)$ and Div are the functions of loss flatness and task discrepancy, respectively. The definitions of δ, θ and c_t^* are the same as (15). Therefore, a desirable solution for continual learning should provide an appropriate stability-plasticity trade-off and an adequate intra/inter-task generalizability, motivating a variety of representative methods.

IV. Method

In this section, we present an elaborated taxonomy of representative continual learning methods (see Fig. 3), analyzing extensively their main motivations, typical implementations and empirical properties. This taxonomy consists of five broad categories that correspond to the major components of a machine learning system (see Fig. 1(c)). These categories have their own theoretical foundations, as analyzed in Section III, while intrinsically connected in objectives. Each category is further divided into several sub-directions depending on their specific implementations.

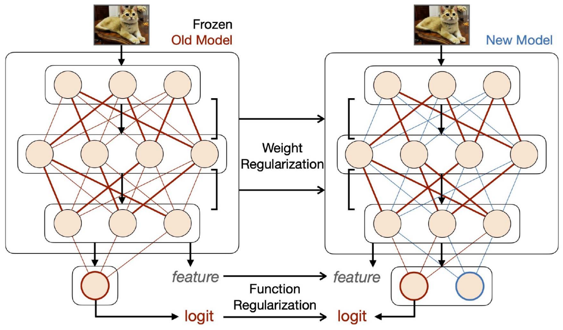


Fig. 4. Regularization-based approach. This direction is characterized by adding explicit regularization terms to mimic the parameters (weight regularization) or behaviors (function regularization) of the old model.

A. Regularization-Based Approach

This direction is characterized by adding explicit regularization terms to balance the old and new tasks, which usually requires storing a frozen copy of the old model for reference (see Fig. 4). Depending on the target of regularization, such methods can be divided into two groups.

The first is *weight regularization*, which selectively regularizes the variation of network parameters. A typical implementation is to add a quadratic penalty in loss function that penalizes the variation of each parameter depending on its "importance" to perform the old tasks (see (5)). The importance can be calculated by the Fisher information matrix (FIM), such as EWC [49] and some more advanced versions [50], [104]. Numerous efforts have been devoted to designing a better importance measurement. SI [105] online approximates the importance of each parameter by its contribution to the total loss variation and its update length over the entire training trajectory. MAS [106] accumulates importance measures based on the sensitivity of predictive results to parameter changes, which is both online and unsupervised. RWalk [44] combines the regularization terms of SI [105] and EWC [49] to integrate their advantages. Interestingly, these importance measurements have been shown to be all tantamount to an approximation of the FIM [107], although stemming from different motivations.

There are also some works focusing on refining the implementation of the quadratic penalty. Since the diagonal approximation of the FIM in EWC [49] might lose information about the old tasks, R-EWC [108] performs a factorized rotation of the parameter space to diagonalize the FIM. XK-FAC [109] further considers the inter-example relations in approximating the FIM to better accommodate batch normalization. Observing the asymmetric effect of parameter changes on old tasks, ALASSO [110] designs an asymmetric quadratic penalty with one of its sides overestimated.

Compared to learning each task within the constraints of the old model, which typically exacerbates the intransience, an *expansion-renormalization* process of obtaining separately the new task solution and renormalizing it with the old model can provide a better stability-plasticity trade-off. IMM [111] is an early attempt that incrementally matches the moment of the posterior distributions for old and new tasks, i.e., a weighted average of their solutions. ResCL [112] extends this idea with a learnable combination coefficient. P&C [104] learns each task individually with an additional network, and then distills it back to the old model with a generalized version of EWC [49]. AFEC [52] introduces a forgetting rate to eliminate the potential negative transfer from the original posterior $p(\theta|\mathcal{D}_{1:k-1})$ in (1), and derives quadratic terms to penalize differences of the network parameters θ with both the old and new task solutions. To reliably average the old and new task solutions, a linear connector [113] is constructed by constraining them on a linear low-error path.

Other forms of regularization that target the network itself also belong to this sub-direction. As discussed before, online variational inference of the posterior distribution can serve as an implicit regularization of parameter changes [55], [56], [58], [59]. Instead of consolidating parameters, NPC [114] estimates the importance of each neuron and selectively reduces its learning rate. UCL [115] and AGS-CL [116] freeze the parameters connecting the important neurons, equivalent to a hard version of weight regularization. The second is *function regularization*, which targets the intermediate or final output of the prediction function. This strategy typically uses the previously-learned model as the teacher and the currently-trained model as the student, while implementing knowledge distillation (KD) [117] to mitigate catastrophic forgetting. Ideally, the target of KD should be all old training samples, which are unavailable in continual learning. The alternatives can be new training samples [118], [119], [120], [121], a small fraction of old training samples [46], [47], [122], [123], external unlabeled data [124], generated data [125], [126], etc., suffering from different degrees of distribution shift.

As a pioneer work, LwF [118] and LwF.MC [122] learn *new training samples* while using their predictions from the output head of the old tasks to compute the distillation loss. LwM [119] exploits the attention maps of new training samples for KD. EBLL [121] learns task-specific autoencoders and prevents changes in feature reconstruction. GD [124] further distills knowledge on the large stream of *unlabeled data* available in the wild. When old training samples are faithfully recovered, the potential of function regularization can be largely released. Thus, function regularization often collaborates with replaying a few *old training samples*, discussed latter in Section IV-B. Meanwhile, sequential Bayesian inference over function space can be seen as a form of function regularization, which generally requires storing some old training samples (called "coreset" in literature), such as FRCL [127], FROMP [128] and S-FSVI [60]. For conditional generation, the *generated data* of previously-learned conditions and their output values are regularized to be consistent between the teacher and student models, such as MeRGANs [125], DRI [129] and LifelongGAN [126].

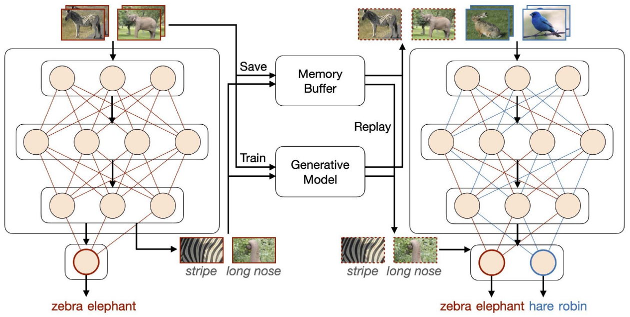


Fig. 5. Replay-based approach. This direction is characterized by approximating and recovering the old data distributions. Typical sub-directions include experience replay, which saves a few old training samples in a memory buffer; generative replay, which trains a generative model to provide generated samples; and feature replay, which recovers the distribution of old features through saving prototypes, saving statistical information or training a generative model.

B. Replay-Based Approach

We group the methods for approximating and recovering old data distributions into this direction (see Fig. 5), which can be further divided into three sub-directions depending on the content of replay, each with its own challenges.

The first is *experience replay*, which typically stores a few old training samples within a small memory buffer. Due to the extremely limited storage space, the key challenges consist of *how to construct* and *how to exploit* the memory buffer. As for construction, the stored training samples should be carefully selected, compressed, augmented, and updated, in order to recover adaptively the past information. Earlier work adopts fixed principles for *sample selection*. For example, Reservoir Sampling [130], [131] randomly stores a fixed amount of training samples from each input batch. Ring Buffer [45] further ensures an equal number of old training samples per class. Mean-of-Feature [122] selects an equal number of old training samples that are closest to the feature mean of each class. More advanced strategies are typically gradient-based or optimizable, by maximizing such as the sample diversity of parameter gradients [21], performance of corresponding tasks with cardinality constraints [132], mini-batch gradient similarity and cross-batch gradient diversity [133], ability of optimizing latent decision boundaries [134], diversity of robustness against perturbations [32], similarity to the gradients of old training samples with respect to the current parameters [135], etc.

To improve *storage efficiency*, AQM [136] performs online continual compression based on a VQ-VAE framework [137] and saves compressed data for replay. MRDC [138] formulates experience replay with data compression as determinantal point processes (DPPs), and derives a computationally efficient way for online determination of the optimal compression rate. RM [32] adopts conventional and label mixing-based strategies of data augmentation to enhance the diversity of old training samples. RAR [139] synthesizes adversarial samples near the forgetting boundary and performs MixUp for data augmentation. The auxiliary information with low storage cost, such as class statistics [79], [140] and attention maps [141], [142], can be further incorporated to maintain old knowledge. Besides, the old training samples can be continually adjusted to accommodate incremental changes, e.g., making them more representative [143] or challenging [144] for separation.

As for exploitation, experience replay requires an adequate use of the memory buffer to recover the past information. There are many different implementations, closely related to other continual learning strategies. First, the effect of old training samples in *optimization* can be constrained to avoid catastrophic forgetting and facilitate knowledge transfer. For example, GEM [45] constructs individual constraints based on the old training samples for each task to ensure non-increase in their losses. A-GEM [63] replaces the individual constraints with a global loss of all tasks to improve training efficiency. LOGD [64] decomposes the gradient of each task into task-sharing and task-specific components to leverage inter-task information. To achieve a good trade-off in interference-transfer [94], [131], MER [131] employs meta-learning for gradient alignment in experience replay. BCL [94] explicitly pursues a saddle point of the cost of old and new training samples. To selectively utilize the memory buffer, MIR [145] prioritizes the old training samples that subject to more forgetting, while HAL [146] uses them as "anchors" and stabilizes their predictions.

On the other hand, experience replay can be naturally combined with *knowledge distillation* (KD), which additionally incorporates the past information from the old model. iCaRL [122] and EEIL [123] are two early works that perform KD on both old and new training samples. Some subsequent improvements focus on different issues in experience replay. To mitigate data imbalance of the limited old training samples, LUCIR [46] encourages similar feature orientation of the old and new models, while performing cosine normalization of the last layer and mining hard negatives of the current task. BiC [147] and WA [148] attribute this issue to the bias of the last fully-connected layer, and resolve it by either learning a bias correction layer with a balanced validation set [147], or normalizing the output weights [148]. SS-IL [149] adopts separated softmax in the last layer and task-wise KD to mitigate the bias. DRI [129] trains a generative model to supplement the old training samples with additional generated data. To alleviate dramatic representation shifts, PODNet [47] employs a spatial distillation loss to preserve representations throughout the model. Co2L [92] introduces a self-supervised distillation loss to obtain robust representations against catastrophic forgetting. GeoDL [150] performs KD along a path that connects the low-dimensional projections of the old and new feature spaces for a smooth transition between them. ELI [151] learns an energy manifold with the old and new models to realign the representation shifts for optimizing incremental tasks. To adequately exploit the past information, DDE [152] distills colliding effects from the features of the new training samples, while CSC [153] additionally leverages the structure of the old feature space. To further enhance learning plasticity, D+R [154] performs KD from an additional model dedicated to each new task. FOSTER [155] expands new modules to fit the residuals of the old model and then distills them into a single model. Besides, weight regularization can also be combined with experience replay for better performance and generality [44], [52].

It is worth noting that the merits and limitations of experience replay remain largely open. In addition to the intuitive benefits of staying in the low-loss region of the old tasks [156], theoretical analysis has demonstrated its contribution to resolving the NP-hard problem of optimal continual learning [62]. However, it risks overfitting to only a few old training samples retained in the memory buffer, which potentially affects generalizability [156]. To alleviate overfitting, LiDER [157] takes inspirations from adversarial robustness and enforces the Lipschitz continuity of the model to its inputs. MOCA [158] enlarges the variation of representations to prevent the old ones from shrinking in their space. Interestingly, several simple baselines of experience replay can achieve considerable performance. DER [31] stores old training samples together with their logits, and perform logit-matching throughout the optimization trajectory. GDumb [159] greedily collects incoming training samples in a memory buffer and then uses them to train a model from scratch for testing. These efforts can serve as evaluation criteria for subsequent exploration.

The second is *generative replay* or pseudo-rehearsal, which generally requires training an additional generative model to replay generated data. This is closely related to continual learning of generative models themselves, as they also require incremental updates. DGR [160] provides an initial framework in which learning each generation task is accompanied with replaying generated data sampled from the old generative model, so as to inherit the previously-learned knowledge. MeRGAN [125] further enforces consistency of the generated data sampled with the same random noise between the old and new generative models, similar to the role of function regularization. Besides, other continual learning strategies can be incorporated into generative replay. Weight regularization [25], [55], [161], [162] and experience replay [25], [163] have been shown to be effective in mitigating catastrophic forgetting of generative models. DGMa/DGMw [164] and a follow-up work [162] adopt binary masks to allocate task-specific parameters for overcoming inter-task interference, and an extendable network to ensure scalability. If pre-training is available, it can provide a relatively stable and strong reference model for continual learning. For example, FearNet [165] and ILCAN [166] additionally preserves statistical information of the old features acquired from a pre-trained feature extractor, while GAN-Memory [167] continually adjusts a pre-trained generative model with task-specific parameters.

The generative models for pseudo-rehearsal can be of various types, such as generative adversarial networks (GANs) and (variational) autoencoder (VAE). A majority of state-of-the-art approaches have focused on GANs to enjoy its advantages in fine-grained generation, but usually suffer from label inconsistency in continual learning [164], [168]. In contrast, autoencoder-based strategies, such as FearNet [165], SRM [169] and EEC [168], can explicitly control the labels of the generated data, although relatively blurred. L-VAEGAN [170] instead employs a hybrid model for both high-quality generation and accurate inference. However, since continual learning of generative models is extremely difficult and requires significant resource overhead, generative replay is typically limited to relatively simple datasets [162], [171]. An alternative is to convert the target of generative replay from data level to feature level, which can largely reduce the complexity of conditional generation and more adequately exploit semantic information. For example, GFR [172] trains a conditional GANs to replay generated features after the feature extractor. BI-R [171] incorporates context-modulated feedback connections in a standard VAE to replay internal representations.

In fact, maintaining feature-level rather than data-level distributions enjoys significant benefits in efficiency and privacy. We categorize this sub-direction as *feature replay*. However, a central challenge is the *representation shift* caused by sequentially updating the feature extractor, which reflects the feature-level catastrophic forgetting. To address this issue, GFR [172], FA [120] and DSR [173] perform feature distillation between the old and new models. IL2M [140] and SNCL [79] recover statistics of feature representations (e.g., mean and covariance) on the basis of experience replay. RER [174] explicitly estimates the representation shift to update the preserved old features. REMIND [175] fixes the early layers of the feature extractor and reconstruct the intermediate representations to update the latter layers. FeTrIL [176] employs a fixed feature extractor learned from the initial task and replays generated features afterwards. For continual learning from scratch, the required changes in representation are often dramatic, while stabilizing the feature extractor may interfere with accommodating new representations. In contrast, the use of strong pre-training can provide robust representations that are generalizable to downstream tasks and remain relatively stable in continual learning, alleviating the central challenge of feature replay.

C. Optimization-Based Approach

Continual learning can be achieved not only by adding additional terms to the loss function (e.g., regularization and replay), but also by explicitly designing and manipulating the optimization programs (see Fig. 6).

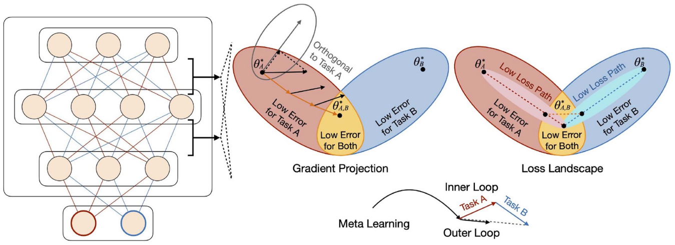


Fig. 6. Optimization-based approach. This direction is characterized by explicit design and manipulation of the optimization programs, such as gradient projection with reference to the gradient space or input space of the old tasks (adapted from [67]), meta-learning of sequentially arrived tasks within the inner loop, and exploitation of mode connectivity and flat minima in loss landscape (adapted from [86], [113]). θ^*_A, θ^*_B and $\theta^*_{A,B}$ are desirable solutions for task A, task B and both of them, respectively.

A typical idea is to perform *gradient projection*. Some replay-based approaches such as GEM [45], A-GEM [63], LOGD [64] and MER [131] constrain parameter updates to align with the direction of experience replay, corresponding to preserving the previous input space and gradient space through a few old training samples. In contrast to saving old training samples, OWM [65] modifies parameter updates to the orthogonal direction of the previous input space. OGD [67] preserves the old gradient directions and rectifies the current gradient directions orthogonal to them. Orthog-Subspace [177] performs continual learning with orthogonal low-rank vector subspaces and keeps the gradients of different tasks orthogonal to each other. GPM [142] maintains the gradient subspace important to the old tasks (i.e., the bases of core gradient space) for orthogonal projection in updating parameters. FS-DGPM [85] dynamically releases unimportant bases of GPM [142] to improve learning plasticity and encourages the convergence to a flat loss landscape. TRGP [178] defines the "trust region" through the norm of gradient projection onto the subspace of previous inputs, so as to selectively reuse the frozen weights of old tasks. Adam-NSCL [66] instead projects candidate parameter updates into the current null space approximated by the uncentered feature covariance of the old tasks, while AdNS [179] considers the shared part of the previous and the current null spaces. NCL [51] unifies Bayesian weight regularization and gradient projection, encouraging parameter updates in the null space of the old tasks while converging to a maximum of the Bayesian approximation posterior. Under the upper bound of the quadratic penalty in Bayesian weight regularization, RGO [180] modifies gradient directions with a recursive optimization procedure to obtain the optimal solution. Therefore, as regularization and replay are ultimately achieved by rectifying the current gradient, gradient projection corresponds to a similar modification but explicitly in parameter updates.

Another attractive idea is *meta-learning* or learning-to-learn for continual learning, which attempts to obtain a data-driven inductive bias for various scenarios, rather than designing it manually [3]. OML [181] provides a meta-training strategy that performs online updates on the sequentially arrived inputs and minimizes their interference, which can naturally obtain sparse representations suitable for continual learning. ANML [182] extends this idea by meta-learning of a context-dependent gating function to selectively activate neurons with respect to incremental tasks. AIM [183] learns a mixture of experts to make predictions with the representations of OML [181] or ANML [182], further sparsifying the representations at the architectural level. Meanwhile, meta-learning can work with experience replay to better utilize both the old and new training samples. For example, MER [131] aligns their gradient directions, while iTAML [184] applies a meta-updating rule to keep them in balance with each other. With the help of experience replay, La-MAML [185] optimizes the OML [181] objective in an online fashion with an adaptively modulated learning rate. OSAKA [43] proposes a hybrid objective of knowledge accumulation and fast adaptation, which can be resolved by obtaining a good initialization with meta-training and then incorporating knowledge of incremental tasks into the initialization. Meta-learning can also be used to optimize specialized architectures. MERLIN [78] consolidates a meta-distribution of model parameters given the representations of each task, which allows to sample task-specific models and ensemble them for inference. Similarly, PR [74] adopts a Bayesian strategy to learn task-specific posteriors with a shared meta-model. MARK [186] maintains a set of shared weights that are incrementally updated with meta-learning and selectively masked to solve specific tasks. ARI [187] combines adversarial attacks with experience replay to obtain task-specific models, which are then fused together through meta-training.

Besides, some other works refine the optimization process from a *loss landscape* perspective. For example, rather than dedicating an algorithm, Stable-SGD [81] enables SGD to find a flat local minima by adapting the factors in training regime, such as dropout, learning rate decay and batch size. MC-SGD [86] empirically demonstrates that the local minima obtained by multi-task learning and continual learning can be connected by a linear path of low error, and applies experience replay to find a solution along it. Linear Connector [113] adopts Adam-NSCL [66] and feature distillation to obtain respective solutions of the old and new tasks connected by a linear path of low error, followed by linear averaging. Further, un-/self-supervised learning (than traditional supervised learning) [28], [93], [188] and large-scale pre-training (than random initialization) [87], [89], [189], [190] have been shown to suffer from less catastrophic forgetting. Empirically, both can be attributed to obtaining a more robust representation [28], [89], [93], [191], and converging to a wider loss basin [28], [87], [93], [192], [193], suggesting a potential link among the sensitivity of representations, parameters and task-specific errors. Many efforts seek to leverage these advantages in continual learning, as we discuss next.

D. Representation-Based Approach

We group the methods that create and exploit the strengths of representations for continual learning into this category (see Fig. 7). In addition to an earlier work on obtaining sparse representations through meta-training [181], recent works have attempted to incorporate the advantages of self-supervised learning (SSL) [91], [93], [188] and large-scale pre-training [87], [189], [191] to improve the representations in initialization and in continual learning. Note that these two strategies are closely related, since the pre-training data is usually of a huge amount and without explicit labels, while the performance of SSL itself is mainly evaluated by fine-tuning on (a sequence of) downstream tasks. Below, we will discuss representative sub-directions.

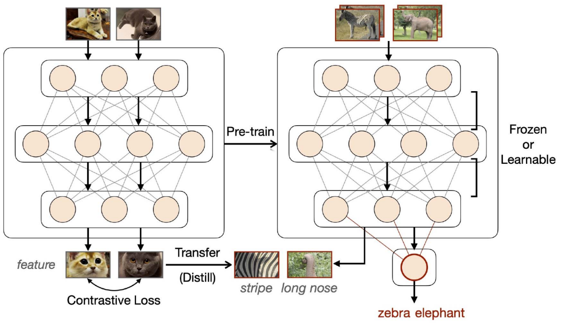


Fig. 7. Representation-based approach. This direction is characterized by creating and leveraging the strengths of representations for continual learning, such as by using self-supervised learning and pre-training.

The first is to implement *self-supervised learning* (basically with contrastive loss) for continual learning. Observing that self-supervised representations are more robust to catastrophic forgetting, LUMP [93] acquires further improvements by interpolating between instances of the old and new tasks. MinRed [194] further promotes the diversity of experience replay by decorrelating the stored old training samples. CaSSLe [195] converts the self-supervised loss to a distillation strategy by mapping the current state of a representation to its previous state. Co2L [92] adopts a supervised contrastive loss to learn each task and a self-supervised loss to distill knowledge between the old and new models. DualNet [91] trains a fast learner with supervised loss and a slow learner with self-supervised loss, with the latter helping the former to acquire generalizable representations. CL-SLAM [196] proposes a dual-network architecture, optimized by self-supervised learning for plasticity and stability, respectively.

The second is to use *pre-training for downstream continual learning*. Several empirical studies suggest that pre-training brings not only strong knowledge transfer but also robustness to catastrophic forgetting to downstream continual learning [87], [89], [188], [197]. In particular, the benefits for downstream continual learning tend to be more apparent when pre-training with larger data size [89], [197], larger model size [89] and contrastive loss [188], [190]. However, a critical challenge is that the pre-trained knowledge needs to be adaptively leveraged for the current task while maintaining generalizability to future tasks. There are various strategies for this problem, depending on whether the pre-trained representations are fixed or not.

As for adapting a *fixed* backbone, Side-Tuning [198] and DLCFT [199] train a lightweight network in parallel with the backbone and fuse their outputs linearly. TwF [200] also trains a sibling network, but distills knowledge from the backbone in a layer-wise manner. GAN-Memory [167] takes advantage of FiLM [201] and AdaFM [202] to learn task-specific parameters for each layer of a pre-trained generative model, while

ADA [203] employs Adapters [204] with knowledge distillation to adjust a pre-trained transformer. Recent prompt-based approaches instruct the representations of a pre-trained transformer with a few prompt parameters. Such methods typically involve construction of task-adaptive prompts and inference of appropriate prompts for testing, by exploring prompt architectures to accommodate task-sharing and task-specific knowledge. Representative strategies include selecting the most relevant prompts from a prompt pool (L2P [205]), optimizing a weighted summation of the prompt pool with attention factors (CODA-Prompt [206]), using explicitly task-sharing and task-specific prompts (DualPrompt [207]) or only task-specific prompts (S-Prompts [208]), progressive expansion of task-specific prompts (Progressive Prompts [209]), etc. Besides, by saving prototypes, appending a nearest class mean (NCM) classifier to the backbone has proved to be a strong baseline [210], [211], which can be further enhanced by transfer learning techniques such as the FiLM adapter [212]. As for optimizing an *updatable* backbone, F2M [84] searches for flat local minima in the pre-training stage, and then learns incremental tasks within the flat region. CwD [191] regularizes the initial-phase representations to be uniformly scattered, which can empirically mimic the representations of joint training. SAM [87], [213] encourages finding a wide basin in downstream continual learning by optimizing the flatness metric. SLCA [214] observes that slowly fine-tuning the backbone of a pre-trained transformer can achieve outstanding performance in continual learning, and further preserves prototype statistics to rectify the output layer.

The third is *continual pre-training* (CPT) or continual meta-training. As the huge amount of data required for pre-training is typically collected in an incremental manner, performing upstream continual learning to improve downstream performance is particularly important. For example, a recent work [215] combines Barlow Twins and EWC to learn representations from incremental unlabeled data. An empirical study finds that self-supervised pre-training is more effective than supervised protocols for continual learning of vision-language models [216], consistent with the results for only visual tasks [28]. Since texts are generally more efficient than images, IncCLIP [217] replays generated hard negative texts conditioned on images and performs multi-modal knowledge distillation for updating CLIP [218]. Meanwhile, continual meta-training needs to address a similar issue that the base knowledge is incrementally enriched and adapted. IDA [219] imposes discriminants of the old and new models to be aligned relative to the old centers, and otherwise leaves the embedding free to accommodate new tasks. ORDER [220] employs unlabeled OOD data with experience replay and feature replay to cope with highly dynamic task distributions.

E. Architecture-Based Approach

The above strategies basically focus on learning all incremental tasks with a shared set of parameters, which is a major cause of inter-task interference. In contrast, constructing task-specific parameters can explicitly resolve this problem. Previous work generally separates this direction into *parameter isolation* and *dynamic architecture*, depending on whether the network architecture is fixed or not. Here, we instead focus on the way of implementing task-specific parameters, extending the above concepts to parameter allocation, model decomposition and modular network (Fig. 8).

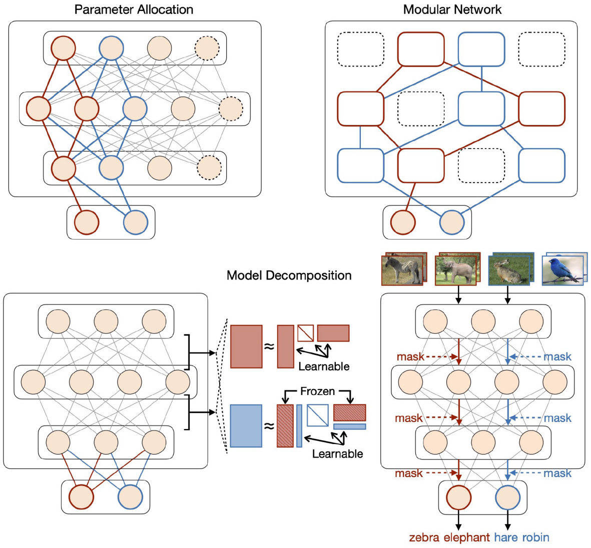


Fig. 8. Architecture-based approach. This direction is characterized by constructing task-specific parameters with a properly-designed architecture, such as assigning dedicated parameters to each task (parameter allocation), constructing task-adaptive sub-modules or sub-networks (modular network), and decomposing the model into task-sharing and task-specific components (model decomposition). Here we exhibit two types of model decomposition, corresponding to parameters (low-rank factorization) and representations (masks of intermediate features).

Parameter allocation features an isolated parameter subspace dedicated to each task throughout the network, where the architecture can be fixed or dynamic in size. Within a *fixed* network architecture, Piggyback [221], HAT [71], WSN [222] and H^2 [223] explicitly optimize a binary mask to select dedicated neurons or parameters for each task, with the masked regions of the old tasks (almost) frozen to prevent catastrophic forgetting. PackNet [224], UCL [115], CLNP [225] and AGS-CL [116] explicitly identify the important neurons or parameters for the current task and then release the unimportant parts to the following tasks, which can be achieved by iterative pruning [224], activation value [116], [225], [226], uncertainty estimation [115], etc. Since the network capacity is limited, “free” parameters tend to saturate as more incremental tasks are introduced. Therefore, these methods typically require sparsity constraints on parameter usage and selective reuse of the frozen old parameters, which might affect the learning of each task. To alleviate this dilemma, the network architecture can be *dynamically expanded* if its capacity is not sufficient to learn a new task well [164], [227], [228]. The dynamic architecture can be explicitly optimized to improve parameter efficiency and knowledge transfer, such as by reinforcement learning [229], [230], architecture search [230], [231], variational Bayes [232], etc. As the network expansion should be much slower than the task increase to ensure scalability, constraints on sparsity and reusability remain important.

Model decomposition explicitly separates a model into task-sharing and task-specific components, where the task-specific components are typically expandable. For a regular network, the task-specific components could be parallel branches [233], [234], adaptive layers [58], [235], masks of intermediate features [186], [236], [237]. Note that the feature masks for model decomposition do not operate in parameter space and are not binary for each task, distinguished from the binary masks for parameter allocation. Besides, the network parameters themselves can be decomposed into task-sharing and task-specific elements, such as by additive decomposition [238], singular value decomposition [239], filter atom decomposition [240] and low-rank factorization [241], [242]. As the number of task-specific components usually grows linearly with incremental tasks, their resource efficiency determines the scalability of this sub-direction.

Modular network leverages parallel sub-networks or sub-modules to learn incremental tasks in a differentiated manner, without pre-defined task-sharing or task-specific components. As an early work, Progressive Networks [70] introduces an identical sub-network for each task and allows knowledge transfer from other sub-networks via adaptor connections. Expert Gate [243] employs a mixture of experts to learn incremental tasks, expanding one expert as each task is introduced. PathNet [72] and RPSNet [244] pre-allocate multiple parallel networks to construct a few candidate paths, i.e., layer-wise compositions of network modules, and select the best path for each task. MNTDP [245] and LMC [246] seek to explicitly find the optimal layout from previous sub-modules and potentially new sub-modules. Similar to parameter allocation, these efforts are intentional to construct task-specific models, while the combination of sub-networks or sub-modules allows explicit reuse of knowledge. In addition, the sub-networks can be encouraged to learn incremental tasks in parallel. Model Zoo [68] expands a sub-network to learn each new task with experience replay of the old tasks, and ensembles all sub-networks for prediction. CoSCL [69] and CAF [103] ensembles multiple continual learning models and modulates the predictive similarity between them, proving to be effective in resolving the discrepancy of task distribution and improving the flatness of loss landscape.

In contrast to other directions, most architecture-based approaches amount to de-correlating incremental tasks in network parameters, which can almost avoid catastrophic forgetting but affect scalability and inter-task generalizability. In particular, task identities are often required to determine which set of parameters to use. To overcome this limitation, task identities can be inferred from the predictive uncertainty of task-specific models [74], [75], [243]. The function of task-identity prediction can also be learned from incremental tasks, using other continual learning strategies to mitigate catastrophic forgetting [78], [223], [233], [241].

V. Application

The real-world complexity presents a variety of particular challenges for continual learning, categorized into *scenario complexity* and *task specificity* (Fig. 1(d)). The former refers to the challenges of continual learning scenarios for each task, such as task-agnostic inference, scarcity of labeled data and generic learning paradigm. We analyze them in Sections V-A, V-B and V-C, respectively, using visual classification as a typical example. The latter refers to the challenges of specific task types. We discuss more complex vision tasks, such as object detection in Section V-D and semantic segmentation in Section V-E, both of which are affected by the co-occurrence of old and new classes in incremental data. The descriptions and task-specific challenges for other domains are left in Appendix A– C, available online, including conditional generation, reinforcement learning, and natural language processing.

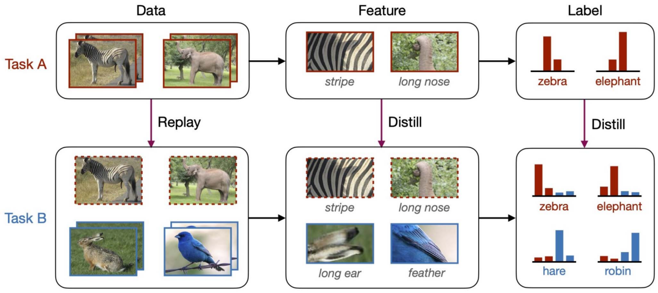


Fig. 9. Representative strategies for class-incremental learning. Catastrophic forgetting can be mitigated with respect to data space (experience replay), feature space (knowledge distillation) and label space (knowledge distillation). This figure is adapted from [152].

A. Task-Agnostic Inference

Continual learning usually has *Task-Incremental Learning* (TIL) as a basic setup, i.e., task identities are provided in both training and testing. In contrast, task-agnostic inference that avoids the use of task identities for testing tends to be more natural but more challenging in practical applications, which is known as *Class-Incremental Learning* (CIL) for classification tasks. For example, let's consider two binary classification tasks: (1) “zebra” and “elephant”; and (2) “hare” and “robin”. After learning these two tasks, TIL needs to know which task it is and then classify the two classes accordingly, while CIL directly classifies the four classes at the same time. Therefore, CIL has received great attention and become almost the most representative scenario for continual learning.

The CIL problem can be disentangled into *within-task prediction* and *task-identity prediction* [75], [77], where the latter is a particular challenge of task-agnostic inference. To cope with CIL, the behavior of the previous model is imposed onto the current model, in terms of data, feature, and label spaces (Fig. 9). As replay of the old training samples can impose an end-to-end effect [152], many state-of-the-art methods are built on the framework of *experience replay* and then incorporate *knowledge distillation*, as discussed in Section IV-B. In Appendix Table 2, available online, we summarize these CIL methods based on their major focuses. To avoid extra resource overhead and potential privacy issues of retaining old training samples, many efforts attempt to perform CIL without experience replay, i.e., *Data-Free* CIL. An intriguing idea is to replay synthetic data produced by inverting a frozen copy of the old classification model [247], [248], which usually further incorporate knowledge distillation to compensate the lost information in model inversion. Other methods exploit the class-wise statistics of feature representations to obtain a balanced classifier, such as by imposing the representations to be transferable and invariant [173], [176], or compensating explicitly the representation shifts [174], [249].

B. Scarcity of Labeled Data

Most of the current continual learning settings assume that incremental tasks have sufficiently large amounts of labeled data, which is often expensive and difficult to obtain in practical applications. For this reason, there is a growing body of work focusing on the scarcity of labeled data in continual learning. A representative scenario is called *Few-Shot* CIL (FSCIL) [24], where the model first learns some base classes for initialization with a large number of training samples, and then learns a sequence of novel classes with only a few training samples. The extremely limited training samples exacerbate the overfitting of previously-learned representations to subsequent tasks, which can be alleviated by recent work such as preserving the topology of representations [24], constructing an exemplar relation graph for knowledge distillation [250], selectively updating only the unimportant parameters [251] or stabilizing the important parameters [252], updating parameters within the flat region of loss landscape [84], meta-learning of a good initialization [253], as well as generative replay [254].

There are many other efforts keeping the initialized backbone fixed in subsequent continual learning, so as to decouple the learning of *representation* and *classifier*. Following this idea, representative strategies can be separated into two aspects. The first is to obtain compatible and extensible representations from massive base classes, such as by enforcing the representations compatible with simulated incremental tasks [255], reserving the feature space with virtual prototypes for future classes [256], using angular penalty loss with data augmentation [257], providing extra constraints from margin-based representations [258], etc. The second is to obtain an adaptive classifier from a sequence of novel classes, such as by evolving the classifier weights with a graph attention network [259], performing hyperdimensional computing [260], sampling stochastic classifiers [261], etc. Besides, auxiliary information such as semantic word vectors [262], [263] and sketch exemplars [264] can be incorporated to enrich the limited training samples.

In addition to a few labeled data, there is usually a large amount of unlabeled data available and collected over time. The first practical setting is called *Semi-Supervised Continual Learning* (SSCL) [25], which considers incremental data as partially labeled. As an initial attempt, ORDisCo [25] learns a semi-supervised classification model together with a conditional GANs for generative replay, and regularizes discriminator consistency to mitigate catastrophic forgetting. Subsequent work includes training an adversarial autoencoder to reconstruct images [265], imposing predictive consistency among augmented and interpolated data [266], and leveraging the nearest-neighbor classifier to distill class-instance relationships [267]. The second scenario assumes that there is an external unlabeled dataset to facilitate supervised continual learning, e.g., by knowledge distillation [124] and data augmentation [268]. The third scenario is to learn representations from incremental unlabeled data [28], [215], [216], which becomes an increasingly important topic for updating pre-trained knowledge in foundation models.

C. Generic Learning Paradigm

Potential challenges of the learning paradigm can be summarized in a broad concept called *General Continual Learning* (GCL) [12], [20], [31], where the model observes incremental data in an online fashion without explicit task boundaries. Correspondingly, GCL consists of two interconnected settings: *Task-Free Continual Learning* (TFCL) [20], where the task identities are not accessible in either training or testing; and *Online Continual Learning* (OCL) [21], where the training samples are observed in an one-pass data stream. Since TFCL usually accesses only a small batch of training samples at a time for gradual changes in task distributions, while OCL usually requires only the data label rather than the task identity for each training sample, many methods for TFCL and OCL are compatible, summarized with their applicable scenarios in Appendix Table 3, available online.

Some of them attempt to learn specialized parameters in a growing architecture. CN-DPM [80] adopts Dirichlet process mixture models to construct a growing number of neural experts, while a concurrent work [269] derives such mixture models from a probabilistic meta-learner. VariGrow [270] employs an energy-based novelty score to decide whether to extend a new expert or update an old one. ODDL [271] estimates the discrepancy between the current memory buffer and the previously-learned knowledge as an expansion signal. InstAParam [272] selects and consolidates appropriate network paths for individual training samples.

On the other hand, many efforts are built on experience replay, focusing on construction, management and exploitation of a memory buffer. Since training samples of the same distribution arrive in small batches, the information of task boundaries is less effective, and reservoir sampling usually serves as an effective baseline strategy for sample selection. More advanced strategies prioritize the replay of those training samples that are informative [273], diversified in parameter gradients [21], balanced in class labels [159], [274], and beneficial to latent decision boundaries [134]. Meanwhile, the memory buffer can be dynamically managed, such as by removing less important training samples [42], editing the old training samples to be more likely forgotten [144], [275], and retrieving the old training samples that are susceptible to interference [145], [276]. To better exploit the memory buffer, representative strategies include calibrating features with task-specific parameters [277], performing knowledge distillation [31], [278], improving representations with contrastive learning [276], [279], using asymmetric cross-entropy [280] or constrained gradient directions [45], [63] of the old and new training samples, repeated rehearsal with data augmentation [281], properly adjusting the learning rate [42], etc.

D. Object Detection

Incremental Object Detection (IOD) is a typical extension of continual learning for object detection, where the training samples annotated with different classes are introduced in sequence, and the model needs to correctly locate and identify the objects belonging to the previously-learned classes. Unlike visual classification with only one object appearing in each training sample, object detection usually has multiple objects belonging to the old and new classes appearing together. Such *co-occurrence* poses an additional challenge for IOD, where the old classes are marked as the background when learning new classes, thus exacerbating catastrophic forgetting. On the other hand, this makes knowledge distillation a naturally powerful strategy for IOD, since the old class objects can be obtained from new training samples to constrain the differences in responses between the old and new models. As an early work, ILOD [282] distills the responses for old classes to prevent catastrophic forgetting on Fast R-CNN. The idea of knowledge distillation is then introduced to other detection frameworks [283], [284], [285]. Some approaches exploit the unlabeled in-the-wild data to distill the old and new models into a shared model, in order to bridge potential non co-occurrence [284] and achieve a better stability-plasticity trade-off [286]. To further improve learning plasticity, IOD-ML [287] adopts meta-learning to reshape parameter gradients into a balanced direction between the old and new classes.

E. Semantic Segmentation

Continual Semantic Segmentation (CSS) aims at pixel-wise prediction of classes and learning new classes in addition to the old ones. Similar to IOD, the old and new classes can appear together with annotations of only the latter, leading to the old classes being treated as the background (known as the *background shift*) and thus exacerbates catastrophic forgetting. A common strategy is to distill knowledge adaptively from the old model, which can faithfully distinguish unannotated old classes from the background. For example, MiB [288] calibrates regular cross-entropy (CE) and knowledge distillation (KD) losses of the background pixels with predictions from the old model. ALIFE [289] further improves the calibrated CE and KD with logit regularization, and fine-tunes the classifier with feature replay. RCIL [290] reparameterizes the network into two parallel branches, where the old branch is frozen for KD between intermediate layers. SDR [291] introduces contrastive learning into distillation of latent representations, where pixels of the same class are clustered and pixels of different classes are separated. PLOP [292], RECALL [293], SSUL [294], Self-Training [295] and WILSON [296] explicitly apply the old model to generate pseudo-labels of the old classes. Auxiliary data resources such as a web crawler [293], a pre-trained generative model [293], unlabeled data [295], and old training samples [294] have been exploited to facilitate KD and prevent catastrophic forgetting. Besides, saliency maps are commonly used to locate unannotated objects in CSS, in response to weak supervision of only image-level annotations [296], as well as defining unknown classes within the background to benefit learning plasticity [294].

VI. Discussion

In this section, we present an in-depth discussion of promising directions for continual learning, including the current trends, essential considerations beyond task performance, and cross-directional prospects.

A. Observation of Current Trend

As continual learning is directly affected by catastrophic forgetting, previous efforts seek to address this critical problem by promoting memory stability of the old knowledge. However, recent work has increasingly focused on facilitating learning plasticity and inter-task generalizability. This essentially advances the understanding of continual learning: a desirable solution requires a proper balance between the old and new tasks, with adequate generalizability to accommodate their distribution differences.

To promote learning plasticity on the basis of memory stability, emergent strategies include renormalization of old and new task solutions [52], [104], [113], [154], [196], balanced exploitation of old and new training samples [46], [129], [147], [148], [149], [152], space reservation for subsequent tasks [115], [116], [256], etc. On the other hand, solution generalizability could be explicitly improved by optimizing the flatness metric [81], [84], [85], [86], [87], constructing an ensemble model at either spatial scale [69], [88] or temporal scale [31], and obtaining well-distributed representations [28], [87], [89], [91], [92], [93]. In particular, since self-supervised and pre-trained representations are naturally more robust to catastrophic forgetting [28], [87], [89], [93], creating, improving and exploiting such representational advantages has become a promising direction.

We also observe that the applications of continual learning are becoming more diverse and widespread. In addition to various scenarios of visual classification, current extensions of continual learning have covered many other visual domains, as well as other related fields such as RL and NLP. We have only introduced some representative applications, with other more specialized and cross-cutting extensions to be explored. Notably, existing work on applications has focused on providing basic benchmarks and baseline approaches. Future work could develop more specialized approaches to obtain stronger performance, or evaluate the generality of current approaches in different applications.

B. Beyond Task Performance

Continual learning can benefit many considerations beyond task performance, such as efficiency, privacy and robustness. A major purpose of continual learning is to avoid retraining all old training samples and thus improve *resource efficiency* of model updates, which is not only applicable to learning multiple incremental tasks, but also important for regular single-task training. Due to the nature of gradient-based optimization, a network tends to “forget” the observed training samples and thus requires repetitive training to capture a distribution, especially for some hard examples [3]. Recent work has shown that the one-pass performance of visual classification can be largely improved by experience replay of hard examples [297] or orthogonal gradient projection [298]. Similarly, resolving within-task catastrophic forgetting can facilitate reinforcement learning [299], [300] and stabilize the training of GANs [301], [302].

Meanwhile, continual learning is relevant to two important directions of *privacy protection*. The first is *Federated Learning* [303], where the server and clients are not allowed to communicate with data. A typical scenario is that the server aggregates the locally trained parameters from multiple clients into a single model and then sends it back. As the incremental data collected by clients is dynamic and variable, federated learning needs to overcome catastrophic forgetting and facilitate knowledge transfer across clients, which can be achieved by continual learning strategies such as model decomposition [304] and knowledge distillation [305], [306]. The second is *Machine Unlearning*, which aims to forget the influence of specific training samples when their access is lost, without affecting other knowledge. Many efforts in this direction are closely related to continual learning, such as learning separate models with subsets of training samples [307], utilizing historical parameters and gradients [308], removing old knowledge from parameters with Fisher information matrix [309], adding adaptive parameters on a pre-trained model [310], etc. On the other hand, continual learning may suffer from data leakage and privacy invasion as retaining all old knowledge. Mnemonic Code [311] embeds a class-specific code when learning each class, enabling to selectively forget them through discarding the corresponding codes. LIRF [312] designs a distillation framework to remove specific old knowledge and store it in a pruned lightweight network for selective recovery.

As a strategy for adapting to variable inputs, continual learning can assist a robust model to eliminate or resist external disturbances [30], [313], [314]. In fact, *robustness* and continual learning are intrinsically linked, as they correspond to generalizability in the spatial and temporal dimensions, respectively. Many ideas for improving robustness to adversarial examples have been used to improve continual learning, such as flat minima [31], [81], model ensemble [69], Lipschitz continuity [157] and adversarial training [158]. Subsequent work could further interconnect excellent ideas from both fields, e.g., designing particular algorithms to actively "forget" [52] external disturbances.

C. Cross-Directional Prospect

Continual learning demonstrates vigorous vitality, as most of the state-of-the-art AI models require flexible and efficient updates, and their advances have contributed to the development of continual learning. Here, we discuss some attractive intersections of continual learning with other topics of the broad AI community:

Diffusion Model [315] is a rising state-of-the-art generative model, which constructs a Markov chain of discrete steps to progressively add random noise for the input and learns to gradually remove the noise to restore the original data distribution. This provides a new target for continual learning, and its outstanding performance in conditional generation can also facilitate the efficacy of generative replay.

Foundation Model, such as GPT [316] and CLIP [218], demonstrates impressive performance in a variety of downstream tasks with the use of large-scale pre-training. The pre-training data is usually huge in volume and collected incrementally, creating urgent demands for efficient updates. On the other hand, an increasing scale of pre-training would facilitate knowledge transfer and mitigate catastrophic forgetting for downstream continual learning. *Transformer-Based Architecture* [317] has proven effective for both language and vision domains, and become the dominant choice for state-of-the-art foundation models. This requires specialized designs to overcome catastrophic forgetting while providing new insights for maintaining task specificity in continual learning. Parameter-efficient fine-tuning techniques originally developed in the field of NLP are being widely adapted to continual learning.

Embodied AI [318] aims to enable AI systems to learn from interactions with the physical environment, rather than static datasets collected primarily from the Internet. The study of general continual learning helps the embodied agents to learn from an egocentric perception similar to humans, and provides a unique opportunity for researchers to pursue the essence of lifelong learning by observing the same person in a long time span. *Advances in Neuroscience* provide important inspirations for the development of continual learning, as biological learning is naturally on a continual basis [1], [3]. The underlying mechanisms include multiple levels from synaptic plasticity to regional collaboration, detailed in Appendix D, available online. With a deeper understanding of the biological brain, more "natural algorithms" can be explored to facilitate continual learning of AI systems.

VII. Conclusion

In this work, we present an up-to-date and comprehensive survey of continual learning, bridging the latest advances in theory, method and application. We summarize both general objectives and particular challenges in this field, with an extensive analysis of how representative strategies address them. Encouragingly, we observe a growing and widespread interest in continual learning from the broad AI community, bringing novel understandings, diversified applications and cross-directional opportunities. Based on such a holistic perspective, we expect the development of continual learning to eventually empower AI systems with human-like adaptability, responding flexibly to real-world dynamics and evolving themselves in a lifelong manner.

References

[1] D. Kudithipudi et al., "Biological underpinnings for lifelong learning machines," *Nat. Mach. Intell.*, vol. 4, pp. 196–210, 2022.

[2] G. I. Parisi et al., "Continual lifelong learning with neural networks: A review," *Neural Netw.*, vol. 113, pp. 54–71, 2019.

[3] R. Hadsell et al., "Embracing change: Continual learning in deep neural networks," *Trends Cogn. Sci.*, vol. 24, pp. 1028–1040, 2020.

[4] G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," 2019, *arXiv:1904.07734*.

[5] Z. Chen and B. Liu, *Lifelong Machine Learning*. San Rafael, CA, USA: Morgan & Claypool, 2018.

[6] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychol. Learn. Motivation*, vol. 24, pp. 109–165, 1989.

[7] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory," *Psychol. Rev.*, vol. 102, pp. 419–457, 1995.

[8] M. Mundt et al., "A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning," *Neural Netw.*, vol. 160, pp. 306–336, 2023.

[9] T. L. Hayes et al., "Replay in deep learning: Current approaches and missing biological elements," *Neural Comput.*, vol. 33, pp. 2908–2950, 2021.

[10] P. Jedlicka et al., "Contributions by metaplasticity to solving the catastrophic forgetting problem," *Trends Neurosci.*, vol. 45, pp. 656–666, 2022.

[11] M. Masana, B. Twardowski, and J. Van de Weijer, "On class orderings for incremental learning," 2020, *arXiv:2007.02145*.

[12] M. De Lange et al., "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, Jul. 2022.

[13] H. Qu et al., "Recent advances of continual learning in computer vision: An overview," 2021, *arXiv:2109.11369*.

[14] Z. Mai et al., "Online continual learning in image classification: An empirical survey," *Neurocomputing*, vol. 469, pp. 28–51, 2022.

[15] M. Biesialska, K. Biesialska, and M. R. Costa-juss à, "Continual lifelong learning in natural language processing: A survey," in *Proc. Int. Conf. Comput. Linguistics*, 2020, pp. 6523–6541.

[16] Z. Ke and B. Liu, "Continual learning of natural language processing tasks: A survey," 2022, *arXiv:2211.12701*.

[17] K. Khetarpal et al., "Towards continual reinforcement learning: A review and perspectives," *J. Artif. Intell. Res.*, vol. 75, pp. 1401–1476, 2022.

[18] V. Lomonaco and D. Maltoni, "Core50: A new dataset and benchmark for continuous object recognition," in *Proc. Conf. Robot Learn.*, 2017, pp. 17–26.

[19] Y.-C. Hsu et al., "Re-evaluating continual learning scenarios: A categorization and case for strong baselines," 2018, *arXiv:1810.12488*.

[20] R. Aljundi, K. Kelchtermans, and T. Tuytelaars, "Task-free continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11246–11255.

[21] R. Aljundi et al., "Gradient based sample selection for online continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, Art. no. 1058.

[22] Y. Sun et al., "ERNIE 2.0: A continual pre-training framework for language understanding," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 8968–8975.

[23] P. Singh, P. Mazumder, P. Rai, and V. P. Namboodiri, "Rectification-based knowledge retention for continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15277–15286.

[24] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12180–12189.

[25] L. Wang, K. Yang, C. Li, L. Hong, Z. Li, and J. Zhu, "ORDisCo: Effective and efficient usage of incremental unlabeled data for semi-supervised continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5379–5388.

[26] K. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Towards open world object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5826–5836.

[27] D. Rao et al., "Continual unsupervised representation learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, Art. no. 687.

[28] D. Hu et al., "How well does self-supervised pre-training perform with streaming data?," in *Proc. Int. Conf. Learn. Representations*, 2021.

[29] C. D. Kim, J. Jeong, and G. Kim, "Imbalanced continual learning with partitioning reservoir sampling," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 411–428.

[30] C. D. Kim, J. Jeong, S. Moon, and G. Kim, "Continual learning on noisy data streams via self-purified replay," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 517–527.

[31] P. Buzzega et al., "Dark experience for general continual learning: A strong, simple baseline," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 1335.

[32] J. Bang, H. Kim, Y. Yoo, J. -W. Ha, and J. Choi, "Rainbow memory: Continual learning with a memory of diverse samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8214–8223.

[33] M. Abdelsalam, M. Faramarzi, S. Sodhani, and S. Chandar, "IIRC: Incremental implicitly-refined classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11033–11042.

[34] M. Liang et al., "Balancing between forgetting and acquisition in incremental subpopulation learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 364–380.

[35] Z. Ke, B. Liu, and X. Huang, "Continual learning of a mixed sequence of similar and dissimilar tasks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 1553.

[36] X. Liu et al., "Long-tailed class incremental learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 495–512.

[37] S. Roy et al., "Class-incremental novel class discovery," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 317–333.

[38] K. Joseph et al., "Novel class discovery without forgetting," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 570–586.

[39] T. Srinivasan et al., "CLiMB: A continual learning benchmark for vision-and-language tasks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 29440–29453.

[40] F. Mi, L. Kong, T. Lin, K. Yu, and B. Faltings, "Generalized class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 970–974.

[41] J. Xie, S. Yan, and X. He, "General incremental learning with domain-aware categorical representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14331–14340.

[42] H. Koh et al., "Online continual learning on class incremental blurry task configuration with anytime inference," in *Proc. Int. Conf. Learn. Representations*, 2021.

[43] M. Caccia et al., "Online fast adaptation and knowledge accumulation (OSAKA): A new approach to continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 1387.

[44] A. Chaudhry et al., "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 556–572.

[45] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6470–6479.

[46] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 831–839.

[47] A. Douillard et al., "PODNet: Pooled outputs distillation for small-tasks incremental learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 86–102.

[48] T. Hastie et al., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Berlin, Germany: Springer, 2009.

[49] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, pp. 3521–3526, 2017.

[50] H. Ritter, A. Botev, and D. Barber, "Online structured laplace approximations for overcoming catastrophic forgetting," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 3742–3752.

[51] T.-C. Kao et al., "Natural continual learning: Success is a journey, not (just) a destination," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 28067–28079.

[52] L. Wang et al., "AFEC: Active forgetting of negative transfer in continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 22379–22391.

[53] F. Huszár, "On quadratic penalties in elastic weight consolidation," 2017, *arXiv:1712.03847*.

[54] J. Martens and R. Grosse, "Optimizing neural networks with kronecker-factored approximate curvature," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2408–2417.

[55] C. V. Nguyen et al., "Variational continual learning," in *Proc. Int. Conf. Learn. Representations*, 2018.

[56] T. Adel, H. Zhao, and R. E. Turner, "Continual learning with adaptive weights (CLAW)," in *Proc. Int. Conf. Learn. Representations*, 2019.

[57] R. Kurle et al., "Continual learning with Bayesian neural networks for non-stationary data," in *Proc. Int. Conf. Learn. Representations*, 2019.

[58] N. Loo, S. Swaroop, and R. E. Turner, "Generalized variational continual learning," in *Proc. Int. Conf. Learn. Representations*, 2020.

[59] S. Kapoor, T. Karaletsos, and T. D. Bui, "Variational auto-regressive Gaussian processes for continual learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5290–5300.

[60] T. G. Rudner et al., "Continual learning via sequential function-space variational inference," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 18871–18887.

[61] H. Tseran et al., "Natural variational continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst. Workshop*, 2018.

[62] J. Knoblauch, H. Husain, and T. Diethe, "Optimal continual learning has perfect memory and is NP-HARD," in *Proc. Int. Conf. Mach. Learn.*, 2020, Art. no. 494.

[63] A. Chaudhry et al., "Efficient lifelong learning with A-GEM," in *Proc. Int. Conf. Learn. Representations*, 2018.

[64] S. Tang, D. Chen, J. Zhu, S. Yu, and W. Ouyang, "Layerwise optimization by gradient decomposition for continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9629–9638.

[65] G. Zeng et al., "Continual learning of context-dependent processing in neural networks," *Nat. Mach. Intell.*, vol. 1, pp. 364–372, 2019.

[66] S. Wang, X. Li, J. Sun, and Z. Xu, "Training networks in null space of feature covariance for continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 184–193.

[67] M. Farajtabar et al., "Orthogonal gradient descent for continual learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 3762–3773.

[68] R. Ramesh and P. Chaudhari, "Model zoo: A growing brain that learns continually," in *Proc. Int. Conf. Learn. Representations*, 2021.

[69] L. Wang et al., "CoSCL: Cooperation of small continual learners is stronger than a big one," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 254–271.

[70] A. A. Rusu et al., "Progressive neural networks," 2016, *arXiv:1606.04671*.

[71] J. Serra et al., "Overcoming catastrophic forgetting with hard attention to the task," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4548–4557.

[72] C. Fernando et al., "PathNet: Evolution channels gradient descent in super neural networks," 2017, *arXiv:1701.08734*.

[73] S. Ebrahimi et al., "Uncertainty-guided continual learning with Bayesian neural networks," in *Proc. Int. Conf. Learn. Representations*, 2019.

[74] C. Henning et al., "Posterior meta-replay for continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 14135–14149.

[75] G. Kim et al., "A theoretical study on solving continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 5065–5079.

[76] F. D'Angelo and C. Henning, "Uncertainty-based out-of-distribution detection requires suitable function space priors," 2021, *arXiv:2110.06020*.

[77] G. Kim et al., "Learnability and algorithm for continual learning," in *Proc. Int. Conf. Mach. Learn.*, 2023, Art. no. 694.

[78] K. J. Joseph and V. N. Balasubramanian, "Meta-consolidation for continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 1205.

[79] Z. Gong et al., "Continual pre-training of language models for math problem understanding with syntax-aware memory network," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 5923–5933.

[80] S. Lee et al., "A neural dirichlet process mixture model for task-free continual learning," in *Proc. Int. Conf. Learn. Representations*, 2019.

[81] S. I. Mirzadeh et al., "Understanding the role of training regimes in continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 613.

[82] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Comput.*, vol. 9, pp. 1–42, 1997.

[83] N. S. Keskar et al., "On large-batch training for deep learning: Generalization gap and sharp minima," in *Proc. Int. Conf. Learn. Representations*, 2017.

[84] G. Shi et al., "Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 6747–6761.

[85] D. Deng et al., "Flattening sharpness for dynamic gradient projection memory benefits continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 18710–18721.

[86] S. I. Mirzadeh et al., "Linear mode connectivity in multitask and continual learning," in *Proc. Int. Conf. Learn. Representations*, 2020.

[87] S. V. Mehta et al., "An empirical investigation of the role of pre-training in lifelong learning," 2021, *arXiv:2112.09153*.

[88] S. Cha et al., "CPR: Classifier-projection regularization for continual learning," in *Proc. Int. Conf. Learn. Representations*, 2020.

[89] V. V. Ramasesh, A. Lewkowycz, and E. Dyer, "Effect of scale on catastrophic forgetting in neural networks," in *Proc. Int. Conf. Learn. Representations*, 2021.

[90] S. I. Mirzadeh et al., "Wide neural networks forget less catastrophically," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 15699–15717.

[91] Q. Pham, C. Liu, and S. Hoi, "DualNet: Continual learning, fast and slow," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 16131–16144.

[92] H. Cha, J. Lee, and J. Shin, "Co2L: Contrastive continual learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9496–9505.

[93] D. Madaan et al., "Representational continuity for unsupervised continual learning," in *Proc. Int. Conf. Learn. Representations*, 2021.

[94] K. Ramakrishnan, R. Panda, Q. Fan, J. Henning, A. Oliva, and R. Feris, "Relationship matters: Relation guided knowledge transfer for incremental learning of object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 1009–1018.

[95] A. Pentina and C. Lampert, "A PAC-Bayesian bound for lifelong learning," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 991–999.

[96] M. A. Bennani, T. Doan, and M. Sugiyama, "Generalisation guarantees for continual learning with orthogonal gradient descent," in *Proc. Int. Conf. Mach. Learn. Workshops*, 2020.

[97] T. Doan et al., "A theoretical analysis of catastrophic forgetting through the NTK overlap matrix," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 1072–1080.

[98] R. Karakida and S. Akaho, "Learning curves for continual learning in neural networks: Self-knowledge transfer and forgetting," in *Proc. Int. Conf. Learn. Representations*, 2022.

[99] S. Lee, S. Goldt, and A. Saxe, "Continual learning in the teacher-student setup: Impact of task similarity," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 6109–6119.

[100] M. Wołczyk et al., "Continual learning with guarantees via weight interval constraints," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 23897–23911.

[101] T. Doan et al., "Efficient continual learning ensembles in neural network subspaces," 2022, *arXiv:2202.09826*.

[102] L. Peng, P. Giampouras, and R. Vidal, "The ideal continual learner: An agent that never forgets," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 27585–27610.

[103] L. Wang et al., "Incorporating neuro-inspired adaptability for continual learning in artificial intelligence," *Nat. Mach. Intell.*, vol. 5, pp. 1356–1368, 2023.

[104] J. Schwarz et al., "Progress & compress: A scalable framework for continual learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4528–4537.

[105] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3987–3995.

[106] R. Aljundi et al., "Memory aware synapses: Learning what (not) to forget," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 139–154.

[107] F. Benzing, "Unifying importance based regularisation methods for continual learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2022, pp. 2372–2396.

[108] X. Liu, M. Masana, L. Herranz, J. Van de Weijer, A. M. López, and A. D. Bagdanov, "Rotate your networks: Better weight consolidation and less catastrophic forgetting," in *Proc. Int. Conf. Pattern Recognit.*, 2018, pp. 2262–2268.

[109] J. Lee, H. G. Hong, D. Joo, and J. Kim, "Continual learning with extended kronecker-factored approximate curvature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8998–9007.

[110] D. Park, S. Hong, B. Han, and K. M. Lee, "Continual learning by asymmetric loss approximation with single-side overestimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3334–3343.

[111] S.-W. Lee et al., "Overcoming catastrophic forgetting by incremental moment matching," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4652–4662.

[112] J. Lee et al., "Residual continual learning," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 4553–4560.

[113] G. Lin, H. Chu, and H. Lai, "Towards better plasticity-stability trade-off in incremental learning: A simple linear connector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 89–98.

[114] I. Paik et al., "Overcoming catastrophic forgetting by neuron-level plasticity control," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 5339–5346.

[115] H. Ahn et al., "Uncertainty-based continual learning with adaptive regularization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 4394–4404.

[116] S. Jung et al., "Continual learning with node-importance based adaptive group sparse regularization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 3647–3658.

[117] J. Gou et al., "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, pp. 1789–1819, 2021.

[118] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[119] P. Dhar, R. V. Singh, K. -C. Peng, Z. Wu, and R. Chellappa, “Learning without memorizing,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5133–5141.

[120] A. Iscen et al., “Memory-efficient incremental learning through feature adaptation,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 699–715.

[121] A. Rannen, R. Aljundi, M. B. Blaschko, and T. Tuytelaars, “Encoder based lifelong learning,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 1329–1337.

[122] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “iCaRL: Incremental classifier and representation learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5533–5542.

[123] F. M. Castro et al., “End-to-end incremental learning,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 233–248.

[124] K. Lee, K. Lee, J. Shin, and H. Lee, “Overcoming catastrophic forgetting with unlabeled data in the wild,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 312–321.

[125] C. Wu et al., “Memory replay GANs: Learning to generate new categories without forgetting,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 5966–5976.

[126] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, “Lifelong GAN: Continual learning for conditional image generation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2759–2768.

[127] M. K. Titsias et al., “Functional regularisation for continual learning with Gaussian processes,” in *Proc. Int. Conf. Learn. Representations*, 2019.

[128] P. Pan et al., “Continual deep learning by functional regularisation of memorable past,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 4453–4464.

[129] Z. Wang et al., “Continual learning through retrieval and imagination,” in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 8594–8602.

[130] A. Chaudhry et al., “On tiny episodic memories in continual learning,” 2019, *arXiv:1902.10486*.

[131] M. Riemer et al., “Learning to learn without forgetting by maximizing transfer and minimizing interference,” in *Proc. Int. Conf. Learn. Representations*, 2018.

[132] Z. Borsos, M. Mutny, and A. Krause, “Coresets via bilevel optimization for continual learning and streaming,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 14879–14890.

[133] J. Yoon et al., “Online coreset selection for rehearsal-based continual learning,” in *Proc. Int. Conf. Learn. Representations*, 2021.

[134] D. Shim et al., “Online class-incremental continual learning with adversarial shapley value,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 9630–9638.

[135] R. Tiwari, K. Killamsetty, R. Iyer, and P. Shenoy, “GCR: Gradient coreset based replay buffer selection for continual learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 99–108.

[136] L. Caccia et al., “Online learned continual compression with adaptive quantization modules,” in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1240–1250.

[137] A. Van Den Oord et al., “Neural discrete representation learning,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6306–6315.

[138] L. Wang et al., “Memory replay with data compression for continual learning,” in *Proc. Int. Conf. Learn. Representations*, 2021.

[139] L. Kumari et al., “Retrospective adversarial replay for continual learning,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 28530–28544.

[140] E. Belouadah and A. Popescu, “IL2M: Class incremental learning with dual memory,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 583–592.

[141] S. Ebrahimi et al., “Remembering for the right reasons: Explanations reduce catastrophic forgetting,” in *Proc. Int. Conf. Learn. Representations*, 2020.

[142] G. Saha, I. Garg, and K. Roy, “Gradient projection memory for continual learning,” in *Proc. Int. Conf. Learn. Representations*, 2020.

[143] Y. Liu, Y. Su, A. -A. Liu, B. Schiele, and Q. Sun, “Mnemonics training: Multi-class incremental learning without forgetting,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12242–12251.

[144] X. Jin et al., “Gradient-based editing of memory examples for online task-free continual learning,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 29193–29205.

[145] R. Aljundi et al., “Online continual learning with maximal interfered retrieval,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 11849–11860.

[146] A. Chaudhry et al., “Using hindsight to anchor past knowledge in continual learning,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 6993–7001.

[147] Y. Wu et al., “Large scale incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 374–382.

[148] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia, “Maintaining discrimination and fairness in class incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13205–13214.

[149] H. Ahn, J. Kwak, S. Lim, H. Bang, H. Kim, and T. Moon, “SS-IL: Separated softmax for incremental learning,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 824–833.

[150] C. Simon, P. Koniusz, and M. Harandi, “On learning the geodesic path for incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1591–1600.

[151] K. Joseph, S. Khan, F. S. Khan, R. M. Anwer, and V. N. Balasubramanian, “Energy-based latent aligner for incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7442–7451.

[152] X. Hu, K. Tang, C. Miao, X. -S. Hua, and H. Zhang, “Distilling causal effect of data in class-incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3956–3965.

[153] A. Ashok, K. Joseph, and V. N. Balasubramanian, “Class-incremental learning with cross-space clustering and controlled transfer,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 105–122.

[154] S. Hou et al., “Lifelong learning via progressive distillation and retrospection,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 437–452.

[155] F.-Y. Wang et al., “Foster: Feature boosting and compression for class-incremental learning,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 398–414.

[156] E. Verwimp, M. De Lange, and T. Tuytelaars, “Rehearsal revealed: The limits and merits of revisiting samples in continual learning,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9365–9374.

[157] L. Bonicelli et al., “On the effectiveness of lipschitz-driven rehearsal in continual learning,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 31886–31901.

[158] L. Yu et al., “Continual learning by modeling intra-class variation,” *Trans. Mach. Learn. Res.*, vol. 2023, 2023.

[159] A. Prabhu, P. H. Torr, and P. K. Dokania, “GDumb: A simple approach that questions our progress in continual learning,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 524–540.

[160] H. Shin et al., “Continual learning with deep generative replay,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 2990–2999.

[161] A. Seff et al., “Continual learning in generative adversarial nets,” 2017, *arXiv:1705.08395*.

[162] L. Wang, B. Lei, Q. Li, H. Su, J. Zhu, and Y. Zhong, “Triple-memory networks: A brain-inspired method for continual learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 5, pp. 1925–1934, May 2022.

[163] C. He et al., “Exemplar-supported generative reproduction for class incremental learning,” in *Proc. Brit. Mach. Vis. Conf.*, 2018.

[164] O. Ostapenko, M. Puscas, T. Klein, P. Jähnichen, and M. Nabi, “Learning to remember: A synaptic plasticity driven framework for continual learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11313–11321.

[165] R. Kemker and C. Kanan, “FearNet: Brain-inspired model for incremental learning,” in *Proc. Int. Conf. Learn. Representations*, 2018.

[166] Y. Xiang, Y. Fu, P. Ji, and H. Huang, “Incremental learning using conditional adversarial networks,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6618–6627.

[167] Y. Cong et al., “GAN memory with no forgetting,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 16481–16494.

[168] A. Ayub and A. Wagner, “EEC: Learning to encode and regenerate images for continual learning,” in *Proc. Int. Conf. Learn. Representations*, 2020.

[169] M. Riemer et al., “Scalable recollections for continual lifelong learning,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1352–1359.

[170] F. Ye and A. G. Bors, “Learning latent representations across multiple data domains using lifelong VAEGAN,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 777–795.

[171] G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias, “Brain-inspired replay for continual learning with artificial neural networks,” *Nat. Commun.*, vol. 11, 2020, Art. no. 4069.

[172] X. Liu et al., “Generative feature replay for class-incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 915–924.

[173] K. Zhu, W. Zhai, Y. Cao, J. Luo, and Z.-J. Zha, “Self-sustaining representation expansion for non-exemplar class-incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9286–9295.

[174] M. Toldo and M. Ozay, “Bring evanescent representations to life in lifelong class incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16711–16720.

[175] T. L. Hayes et al., “Remind your neural network to prevent catastrophic forgetting,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 466–483.

[176] G. Petit, A. Popescu, H. Schindler, D. Picard, and B. Delezoide, "FeTrIL: Feature translation for exemplar-free class-incremental learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 3900–3909.

[177] A. Chaudhry et al., "Continual learning in low-rank orthogonal subspaces," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 9900–9911.

[178] S. Lin et al., "TRGP: Trust region gradient projection for continual learning," in *Proc. Int. Conf. Learn. Representations*, 2021.

[179] Y. Kong et al., "Balancing stability and plasticity through advanced null space in continual learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 219–236.

[180] H. Liu and H. Liu, "Continual learning with recursive gradient optimization," in *Proc. Int. Conf. Learn. Representations*, 2021.

[181] K. Javed and M. White, "Meta-learning representations for continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 1818–1828.

[182] S. Beaulieu et al., "Learning to continually learn," in *Proc. Eur. Conf. Artif. Intell.*, 2020, pp. 992–1001.

[183] E. Lee, C.-H. Huang, and C.-Y. Lee, "Few-shot and continual learning with attentive independent mechanisms," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9435–9444.

[184] J. Rajasegaran, S. Khan, M. Hayat, F. S. Khan, and M. Shah, "iTAML: An incremental task-agnostic meta-learning approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13585–13594.

[185] G. Gupta, K. Yadav, and L. Paull, "Look-ahead meta learning for continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 11588–11598.

[186] J. Hurtado, A. Raymond, and A. Soto, "Optimizing reusable knowledge for continual learning via metalearning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 14150–14162.

[187] R. Wang et al., "Anti-retroactive interference for lifelong learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 163–178.

[188] J. Gallardo, T. L. Hayes, and C. Kanan, "Self-supervised training enhances online continual learning," in *Proc. Brit. Mach. Vis. Conf.*, 2021.

[189] T.-Y. Wu et al., "Class-incremental learning with strong pre-trained models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9591–9600.

[190] M. Davari, N. Asadi, S. Mudur, R. Aljundi, and E. Belilovsky, "Probing representation forgetting in supervised and unsupervised continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16691–16700.

[191] Y. Shi et al., "Mimicking the oracle: An initial phase decorrelation approach for class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16701–16710.

[192] B. Neyshabur, H. Sedghi, and C. Zhang, "What is being transferred in transfer learning?," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 512–523.

[193] Y. Hao et al., "Visualizing and understanding the effectiveness of BERT," in *Proc. Conf. Empir. Methods Natural Lang. Process. Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4141–4150.

[194] S. Purushwalkam, P. Morgado, and A. Gupta, "The challenges of continuous self-supervised learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 702–721.

[195] E. Fini, V. G. T. Da Costa, X. Alameda-Pineda, E. Ricci, K. Alahari, and J. Mairal, "Self-supervised models are continual learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9611–9620.

[196] N. Vödisch et al., "Continual SLAM: Beyond lifelong simultaneous localization and mapping through continual learning," in *Proc. Int. Symp. Robot. Res.*, 2022, pp. 19–35.

[197] O. Ostapenko et al., "Foundational models for continual learning: An empirical study of latent replay," in *Proc. Conf. Lifelong Learn. Agents*, 2022, pp. 534–547.

[198] J. Zhang et al., "Side-tuning: A baseline for network adaptation via additive side networks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 698–714.

[199] H. Shon et al., "DLCFT: Deep linear continual fine-tuning for general incremental learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 513–529.

[200] M. Boschini et al., "Transfer without forgetting," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 692–709.

[201] E. Perez et al., "FiLM: Visual reasoning with a general conditioning layer," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3942–3951.

[202] M. Zhao, Y. Cong, and L. Carin, "On leveraging pretrained GANs for generation with limited data," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 11340–11351.

[203] B. Ermis et al., "Memory efficient continual learning with transformers," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 10629–10642.

[204] N. Houlsby et al., "Parameter-efficient transfer learning for NLP," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2790–2799.

[205] Z. Wang et al., "Learning to prompt for continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 139–149.

[206] J. S. Smith et al., "CODA-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 11909–11919.

[207] Z. Wang et al., "DualPrompt: Complementary prompting for rehearsal-free continual learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 631–648.

[208] Y. Wang, Z. Huang, and X. Hong, "S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 5682–5695.

[209] A. Razdaibiedina et al., "Progressive prompts: Continual learning for language models," in *Proc. Int. Conf. Learn. Representations*, 2023.

[210] P. Janson et al., "A simple baseline that questions the use of pretrained-models in continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst. Workshop*, 2017.

[211] F. Pelosin, "Simpler is better: Off-the-shelf continual learning through pretrained backbones," 2022, *arXiv:2205.01586*.

[212] A. Panos et al., "First session adaptation: A strong replay-free baseline for class-incremental learning," 2023, *arXiv:2303.13199*.

[213] P. Foret et al., "Sharpness-aware minimization for efficiently improving generalization," in *Proc. Int. Conf. Learn. Representations*, 2020.

[214] G. Zhang et al., "SLCA: Slow learner with classifier alignment for continual learning on a pre-trained model," 2023, *arXiv:2303.05118*.

[215] V. Marsocci and S. Scardapane, "Continual barlow twins: Continual self-supervised learning for remote sensing semantic segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 5049–5060, 2023.

[216] A. Cossu et al., "Continual pre-training mitigates forgetting in language and vision," 2022, *arXiv:2205.09357*.

[217] S. Yan et al., "Generative negative text replay for continual vision-language pretraining," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 22–38.

[218] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[219] Q. Liu et al., "Incremental meta-learning via indirect discriminant alignment," 2020, *arXiv:2002.04162*.

[220] Z. Wang et al., "Meta-learning with less forgetting on large-scale non-stationary task distributions," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 221–238.

[221] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 67–82.

[222] H. Kang et al., "Forget-free continual learning with winning subnetworks," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 10734–10750.

[223] H. Jin and E. Kim, "Helpful or harmful: Inter-task association in continual learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 519–535.

[224] A. Mallya and S. Lazebnik, "PackNet: Adding multiple tasks to a single network by iterative pruning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7765–7773.

[225] S. Golkar, M. Kagan, and K. Cho, "Continual learning via neural pruning," 2019, *arXiv:1903.04476*.

[226] M. B. Gurbuz and C. Dovrolis, "NISPA: Neuro-inspired stability-plasticity adaptation for continual learning in sparse networks," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 8157–8174.

[227] J. Yoon et al., "Lifelong learning with dynamically expandable networks," in *Proc. Int. Conf. Learn. Representations*, 2018.

[228] C.-Y. Hung et al., "Compacting, picking and growing for unforgetting continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 13647–13657.

[229] J. Xu and Z. Zhu, "Reinforced continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 907–916.

[230] Q. Qin et al., "BNS: Building network structures dynamically for continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 20608–20620.

[231] X. Li et al., "Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3925–3934.

[232] A. Kumar, S. Chatterjee, and P. Rai, "Bayesian structural adaptation for continual learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5850–5860.

[233] S. Ebrahimi et al., “Adversarial continual learning,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 386–402.

[234] Z. Wu, C. Baek, C. You, and Y. Ma, “Incremental learning via rate reduction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1125–1133.

[235] A. Douillard, A. Ramé, G. Couairon, and M. Cord, “DyTox: Transformers for continual learning with dynamic token expansion,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9275–9285.

[236] P. Singh et al., “Calibrating CNNs for lifelong learning,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 1307.

[237] D. Abati, J. Tomczak, T. Blankevoort, S. Calderara, R. Cucchiara, and B. E. Bejnordi, “Conditional channel gated networks for task-aware continual learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3930–3939.

[238] J. Yoon et al., “Scalable and order-robust continual learning with additive parameter decomposition,” in *Proc. Int. Conf. Learn. Representations*, 2019.

[239] M. Kanakis et al., “Reparameterizing convolutions for incremental multi-task learning without task interference,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 689–707.

[240] Z. Miao et al., “Continual learning with filter atom swapping,” in *Proc. Int. Conf. Learn. Representations*, 2021.

[241] N. Mehta et al., “Continual learning using a Bayesian nonparametric dictionary of weight factors,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 100–108.

[242] R. Hyder et al., “Incremental task learning with incremental rank updates,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 566–582.

[243] R. Aljundi, P. Chakravarty, and T. Tuytelaars, “Expert gate: Lifelong learning with a network of experts,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7120–7129.

[244] J. Rajasegaran et al., “Random path selection for incremental learning,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 12669–12679.

[245] T. Veniat, L. Denoyer, and M. Ranzato, “Efficient continual learning with modular networks and task-driven priors,” in *Proc. Int. Conf. Learn. Representations*, 2020.

[246] O. Ostapenko et al., “Continual learning via local module composition,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 30298–30312.

[247] H. Yin et al., “Dreaming to distill: Data-free knowledge transfer via deepinversion,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8712–8721.

[248] J. Smith, Y. -C. Hsu, J. Balloch, Y. Shen, H. Jin, and Z. Kira, “Always be dreaming: A new approach for data-free class-incremental learning,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9354–9364.

[249] L. Yu et al., “Semantic drift compensation for class-incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6980–6989.

[250] S. Dong et al., “Few-shot class-incremental learning via relation knowledge distillation,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1255–1263.

[251] P. Mazumder, P. Singh, and P. Rai, “Few-shot lifelong learning,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2337–2345.

[252] A. Kukleva, H. Kuehne, and B. Schiele, “Generalized and incremental few-shot learning by explicit learning and calibration without forgetting,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9000–9009.

[253] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, and J. Tang, “MetaFSCIL: A meta-learning approach for few-shot class incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14146–14155.

[254] H. Liu et al., “Few-shot class-incremental learning via entropy-regularized data-free replay,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 146–162.

[255] K. Zhu, Y. Cao, W. Zhai, J. Cheng, and Z.-J. Zha, “Self-promoted prototype refinement for few-shot class-incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6797–6806.

[256] D.-W. Zhou, F. -Y. Wang, H.-J. Ye, L. Ma, S. Pu, and D.-C. Zhan, “Forward compatible few-shot class-incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9036–9046.

[257] C. Peng et al., “Few-shot class-incremental learning from an open-set perspective,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 382–397.

[258] Y. Zou et al., “Margin-based few-shot class-incremental learning with class-level overfitting mitigation,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 27267–27279.

[259] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, and Y. Xu, “Few-shot incremental learning with continually evolved classifiers,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12450–12459.

[260] M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi, “Constrained few-shot class-incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9047–9057.

[261] J. Kalla and S. Biswas, “S3C: Self-supervised stochastic classifiers for few-shot class-incremental learning,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 432–448.

[262] A. Cheraghian, S. Rahman, P. Fang, S. K. Roy, L. Petersson, and M. Harandi, “Semantic-aware knowledge distillation for few-shot class-incremental learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2534–2543.

[263] A. F. Akyürek et al., “Subspace regularizers for few-shot class incremental learning,” in *Proc. Int. Conf. Learn. Representations*, 2021.

[264] A. K. Bhunia et al., “Doodle it yourself: Class incremental learning by drawing a few sketches,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2283–2292.

[265] A. Lechat, S. Herbin, and F. Jurie, “Semi-supervised class incremental learning,” in *Proc. Int. Conf. Pattern Recognit.*, 2021, pp. 10383–10389.

[266] M. Boschini et al., “Continual semi-supervised learning through contrastive interpolation consistency,” *Pattern Recognit. Lett.*, vol. 162, pp. 9–14, 2022.

[267] Z. Kang, E. Fini, M. Nabi, E. Ricci, and K. Alahari, “A soft nearest-neighbor framework for continual semi-supervised learning,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 11834–11843.

[268] Y.-M. Tang, Y.-X. Peng, and W.-S. Zheng, “Learning to imagine: Diversify memory for incremental learning using unlabeled data,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9539–9548.

[269] G. Jerfel et al., “Reconciling meta-learning and continual learning with online mixtures of tasks,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 9119–9130.

[270] R. Ardywibowo et al., “VariGrow: Variational architecture growing for task-agnostic continual learning based on Bayesian novelty,” in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 865–877.

[271] F. Ye and A. G. Bors, “Task-free continual learning via online discrepancy distance learning,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 23675–23688.

[272] H.-J. Chen et al., “Mitigating forgetting in online continual learning via instance-aware parameterization,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 17466–17477.

[273] S. Sun et al., “Information-theoretic online memory selection for continual learning,” in *Proc. Int. Conf. Learn. Representations*, 2021.

[274] M. De Lange and T. Tuytelaars, “Continual prototype evolution: Learning online from non-stationary data streams,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8230–8239.

[275] Z. Wang et al., “Improving task-free continual learning by distributionally robust memory evolution,” in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 22985–22998.

[276] Y. Gu, X. Yang, K. Wei, and C. Deng, “Not just selection, but exploration: Online class-incremental continual learning via dual view consistency,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7432–7441.

[277] Q. Pham et al., “Contextual transformation networks for online continual learning,” in *Proc. Int. Conf. Learn. Representations*, 2020.

[278] J. He, R. Mao, Z. Shao, and F. Zhu, “Incremental learning in online scenario,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13923–13932.

[279] Z. Mai, R. Li, H. Kim, and S. Sanner, “Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3584–3594.

[280] L. Caccia et al., “New insights on reducing abrupt representation change in online continual learning,” in *Proc. Int. Conf. Learn. Representations*, 2021.

[281] Y. Zhang et al., “A simple but strong baseline for online continual learning: Repeated augmented rehearsal,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 14771–14783.

[282] K. Shmelkov, C. Schmid, and K. Alahari, “Incremental learning of object detectors without catastrophic forgetting,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 3420–3429.

[283] C. Peng, K. Zhao, and B. C. Lovell, “Faster ILOD: Incremental learning for object detectors based on faster RCNN,” *Pattern Recognit. Lett.*, vol. 140, pp. 109–115, 2020.

[284] N. Dong et al., “Bridging non co-occurrence with unlabeled in-the-wild data for incremental object detection,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 30492–30503.

[285] T. Feng, M. Wang, and H. Yuan, "Overcoming catastrophic forgetting in incremental object detection via elastic response distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9417–9426.

[286] J. Zhang et al., "Class-incremental learning via deep model consolidation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 1120–1129.

[287] K. Joseph, J. Rajasegaran, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Incremental object detection via meta-learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9209–9216, Dec. 2022.

[288] F. Cermelli, M. Mancini, S. Rota Bulò, E. Ricci, and B. Caputo, "Modeling the background for incremental learning in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9230–9239.

[289] Y. Oh, D. Baek, and B. Ham, "ALIFE: Adaptive logit regularizer and feature replay for incremental semantic segmentation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 14516–14528.

[290] C.-B. Zhang, J. -W. Xiao, X. Liu, Y. -C. Chen, and M.-M. Cheng, "Representation compensation networks for continual semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7043–7054.

[291] U. Michieli and P. Zanuttigh, "Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1114–1124.

[292] A. Douillard, Y. Chen, A. Dapogny, and M. Cord, "PLOP: Learning without forgetting for continual semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4039–4049.

[293] A. Maracani, U. Michieli, M. Toldo, and P. Zanuttigh, "RECALL: Replay-based continual learning in semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7006–7015.

[294] S. Cha et al., "SSUL: Semantic segmentation with unknown label for exemplar-based class-incremental learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 10919–10930.

[295] L. Yu, X. Liu, and J. Van de Weijer, "Self-training for class-incremental semantic segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 9116–9127, Nov. 2023.

[296] F. Cermelli, D. Fontanel, A. Tavera, M. Ciccone, and B. Caputo, "Incremental learning in semantic segmentation from image labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4361–4371.

[297] H. Hu et al., "One pass ImageNet," in *Proc. Int. Conf. Neural Inf. Process. Syst. Workshop*, 2021.

[298] Y. Min, K. Ahn, and N. Azizan, "One-pass learning via bridging orthogonal gradient descent and recursive least-squares," in *Proc. IEEE Conf. Decis. Control*, 2022, pp. 4720–4725.

[299] C. Kaplanis, M. Shanahan, and C. Clopath, "Continual reinforcement learning with complex synapses," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2497–2506.

[300] C. Kaplanis, M. Shanahan, and C. Clopath, "Policy consolidation for continual reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3242–3251.

[301] H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in GANs," in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 1–10.

[302] K. S. Lee, N.-T. Tran, and N.-M. Cheung, "InfoMax-GAN: Improved adversarial image generation via information maximization and contrastive learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 3941–3951.

[303] B. McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[304] J. Yoon et al., "Federated continual learning with weighted inter-client transfer," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12073–12086.

[305] J. Dong et al., "Federated class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10154–10163.

[306] Y. Ma et al., "Continual federated learning based on knowledge distillation," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2022, pp. 2182–2188.

[307] L. Bourtoule et al., "Machine unlearning," in *Proc. Symp. Secur. Privacy*, 2021, pp. 141–159.

[308] Y. Wu, E. Dobriban, and S. Davidson, "DeltaGrad: Rapid retraining of machine learning models," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10355–10366.

[309] A. Golatkar, A. Achille, and S. Soatto, "Eternal sunshine of the spotless net: Selective forgetting in deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9301–9309.

[310] A. Golatkar, A. Achille, A. Ravichandran, M. Polito, and S. Soatto, "Mixed-privacy forgetting in deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 792–801.

[311] T. Shibata et al., "Learning with selective forgetting," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 989–996.

[312] J. Ye et al., "Learning with recoverable forgetting," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 87–103.

[313] M. Zhou et al., "Image de-raining via continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4905–4914.

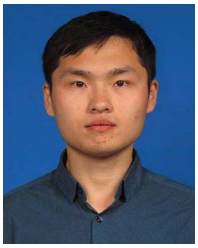
[314] M. Rostami, L. Spinoulas, M. Hussein, J. Mathai, and W. Abd-Almageed, "Detection and continual learning of novel face presentation attacks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 14831–14840.

[315] Y. Song et al., "Score-based generative modeling through stochastic differential equations," in *Proc. Int. Conf. Learn. Representations*, 2020.

[316] T. Brown et al., "Language models are few-shot learners," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1877–1901.

[317] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[318] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied AI: From simulators to research tasks," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 2, pp. 230–244, Apr. 2022.



Liyuan Wang received the BS and PhD degrees from Tsinghua University. He is currently a postdoc with Tsinghua University, working with Prof. Jun Zhu with the Department of Computer Science and Technology. His research interests include continual learning, incremental learning, lifelong learning and brain-inspired AI. His work in continual learning has been published in major conferences and journals in related fields, such as *Nature Machine Intelligence*, NeurIPS, ICLR, CVPR, ICCV, ECCV, etc.

Xingxing Zhang received the BE and PhD degrees from the Institute of Information Science, Beijing Jiaotong University, in 2015 and 2020, respectively. She was also a visiting student with the Department of Computer Science, University of Rochester, from 2018 to 2019. She was a postdoc with the Department of Computer Science and Technology, Tsinghua University, from 2020 to 2022. Her research interests include continual learning and zero/few-shot learning. She has received the excellent PhD thesis award from the Chinese Institute of Electronics, in 2020.

Hang Su (Member, IEEE) is an associated professor with the Department of Computer Science and Technology, Tsinghua University. His research interests lie in the adversarial machine learning and robust computer vision, based on which he has published more than 50 papers including CVPR, ECCV, *IEEE Transactions on Medical Imaging*, etc. He has served as area chair in NeurIPS and the workshop co-chair in AAAI22. He received "Young Investigator Award" from MICCAI2012, the "Best Paper Award" in AVSS2012, and "Platinum Best Paper Award" in ICME2018.

Jun Zhu (Fellow, IEEE) received the BS and PhD degrees from the Department of Computer Science and Technology, Tsinghua University, where he is currently a Bosch AI professor. He was a postdoctoral fellow and adjunct faculty with the Machine Learning Department, Carnegie Mellon University. His research interest is primarily on developing machine learning methods to understand scientific and engineering data arising from various fields. He regularly serves as senior area chairs and area chairs with prestigious conferences, including ICML, NeurIPS, ICLR, IJCAI and AAAI. He was selected as "AI's 10 to Watch" by IEEE Intelligent Systems. He is a fellow of AAAI, and an associate editor-in-chief of *IEEE Transactions on Pattern Analysis and Machine Intelligence*.