Self-Regulated Neurogenesis for Online Data-Incremental Learning

Murat Onur Yildirim[1], **Elif Ceren Gok Yildirim**[1], **Decebal Constantin Mocanu**[2], **Joaquin Vanschoren**[1]
[1]Eindhoven University of Technology, [2]University of Luxembourg

Abstract

Neural networks often struggle with catastrophic forgetting when learning sequences of tasks or data streams, unlike humans who can continuously learn and consolidate new concepts even in the absence of explicit cues. Online data-incremental learning seeks to emulate this capability by processing each sample only once, without having access to task or stream cues at any point in time since this is more realistic compared to offline setups, where all data from novel class(es) is assumed to be readily available. However, existing methods typically rely on storing the subsets of data in memory or expanding the initial model architecture, resulting in significant computational overhead. Drawing inspiration from 'self-regulated neurogenesis'—brain's mechanism for creating specialized regions or circuits for distinct functions—we propose a novel approach SERENA which encodes each concept in a specialized network path called 'concept cell', integrated into a single over-parameterized network. Once a concept is learned, its corresponding concept cell is frozen, effectively preventing the forgetting of previously acquired information. Furthermore, we introduce two new continual learning scenarios that more closely reflect real-world conditions, characterized by gradually changing sample sizes. Experimental results show that our method not only establishes new state-of-the-art results across ten benchmarks but also remarkably surpasses offline supervised batch learning performance. The code is available at https://github.com/muratonuryildirim/serena.

1 Introduction

Artificial neural networks have achieved great success, often surpassing human performance in various applications, particularly when operating on datasets assumed to be sampled from independent and identically distributed (iid) static sources. However, real-world systems face a different reality, where they are exposed to sequential streams of non-stationary (non-iid) data. As the input data changes, previously learned weights are overwritten, leading to *catastrophic forgetting* (McCloskey & Cohen, 1989) which poses a major obstacle to effective continual learning (CL) of sequential streams.

In contrast, humans excel at CL by seamlessly assimilating and integrating novel information from a dynamic and continuous stream of diverse experiences throughout life's continuum. This ability is underpinned by a sophisticated set of neurophysiological processing principles, also called *plasticity-stability* mechanisms (Parisi et al., 2019; Goyal & Bengio, 2022). By continuously monitoring sensory inputs and internal representations, the human brain detects deviations from expected patterns or environmental contexts, signaling concept drift (Zou et al., 2020; Tsuda et al., 2020). In response, it triggers adaptive mechanisms such as updating specialized regions known as *concept cells* (Quiroga, 2012), reallocating attentional resources, or initiating new learning processes to accommodate an evolving environment (Lourenco & Casey, 2013; Astle & Scerif, 2011). Collectively, these processes contribute to *self-regulated neurogenesis* (Kempermann et al., 1998; Yiu & He, 2006), enabling continuous adaptation and knowledge refinement.

Inspired by these intricate mechanisms, we introduce SERENA, a model that learns each concept within a specialized neural circuit (or network path), integrated into a single over-parameterized network. It does not require multiple epochs of training and explicit task cues by automatically detecting concept changes in the data stream thereby can operate in online data-incremental learning scenarios (De Lange & Tuytelaars, 2021)—also referred to as task-free (Aljundi et al., 2019b) or task-agnostic continual learning (Zeno et al., 2019). Once a concept is learned, its corresponding concept cell is frozen to prevent forgetting. During inference, rather than simply relying on a single neural path, SERENA adopts another neuro-inspired strategy that incorporates a built-in recency effect (Murdock Jr, 1962) with an ensemble approach. This prioritizes the recent experiences in the decision-making process while still integrating past knowledge, enhancing adaptability to evolving data.

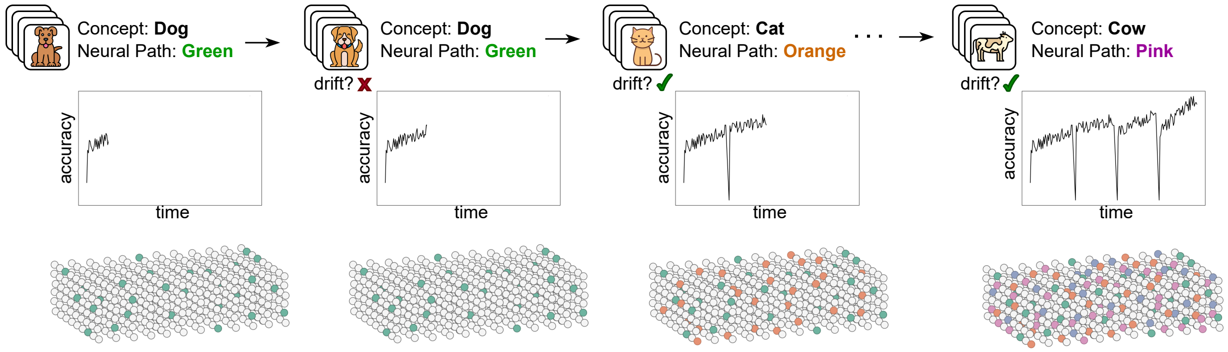


Figure 1: SERENA assigns random network paths or concept cells (Quiroga, 2012) integrated into a single over-parameterized network at each concept drift that is detected automatically without explicit indicators, mirroring the self-regulated neurogenesis (Kempermann et al., 1998; Yiu & He, 2006) without any model growth.

Our contributions can be summarized as follows:

I. **SERENA:** We introduce a novel online data-incremental learning approach inspired by the brain's mechanisms, leveraging concept cells formed through zero-cost random pruning during training and a recency-biased ensemble strategy during inference. It offers a simple yet effective solution by continually fine-tuning specific neural paths within a fixed backbone without relying on experience replay or network expansion.

II. **New SOTA:** We conduct extensive experiments across eleven benchmarks, demonstrating that SERENA significantly outperforms existing state-of-the-art methods while reducing complexity. Notably, it improves the accuracy of state-of-the-art methods more than 2× on Split-CIFAR100 while even surpassing the performance of traditional iid offline supervised batch learning.

III. **New CL Scenarios:** We also propose two new continual learning scenarios that more closely reflect real-world dynamics with gradually increasing or decreasing sample sizes throughout learning sessions, complementing the long-tailed CL scenario (Liu et al., 2022b) and offering a challenging testbed for evaluating continual learning algorithms.

2 Related Work

Continual Learning. Continual learning approaches primarily target task-incremental learning (TIL) or class-incremental learning (CIL) settings, where models are typically aware of task identifiers during training, testing, or both. These approaches often assume that models can be trained over multiple epochs with repeated shuffling. Regularization-based techniques focus on consolidation (Kirkpatrick et al., 2017; Zenke et al., 2017; Chaudhry et al., 2018; Aljundi et al., 2018; Lee et al., 2020a) or knowledge distillation (Li & Hoiem, 2017; Hou et al., 2018; Kang et al., 2022b) to maintain model stability. Replay-based methods either store (Rebuffi et al., 2017; Lopez-Paz & Ranzato, 2017; Wu et al., 2019; Zhao et al., 2020; Arani et al., 2022; Sarfraz et al., 2023) or generate (Mocanu et al., 2016b; Shin et al., 2017; He et al., 2018; Hu et al., 2019; Petit et al., 2023) samples to preserve past experiences. Architecture expansion methods (Aljundi et al., 2017; Liu et al., 2021; Pham et al., 2021; Yan et al., 2021; Wang et al., 2022a; 2023; Douillard et al., 2022; Zhou et al., 2023) add completely new layers or networks to enhance model plasticity and parameter isolation techniques use mask learning (Mallya et al., 2018), iterative pruning (Mallya & Lazebnik, 2018; Golkar et al., 2019; Wortsman et al., 2020; Dekhovich et al., 2023), or dynamic sparse training (Sokar et al., 2021; Gurbuz & Dovrolis, 2022; Kang et al., 2022a; Wang et al., 2022b; Sokar et al., 2023; Kang et al., 2023; Hu et al., 2024) to retain critical parameters. Recent research has integrated parameter-efficient fine-tuning (PEFT) techniques into continual learning to reduce task interference by training only small sets of task-specific parameters on top of the large pre-trained backbone (Zhou et al., 2024; Wang et al., 2022c; Smith et al., 2023; Yildirim et al., 2025). Despite their success, these methods often face practical limitations such as the need for explicit task identifiers, the computational burden of multiple training epochs or pre-trained architectures, which underscores more efficient strategies.

Online Data-Incremental Learning. A more realistic direction argues that realistic continual learners should be capable of processing novel data streams in the absence of explicit task cues at any time point. This concept was first explored in (Aljundi et al., 2019b) and later extended by MIR (Aljundi et al., 2019a), which enhanced reservoir sampling with a loss-based retrieval strategy. Reservoir (Vitter, 1985) serves as a replay baseline with a strong potential to surpass continual learning methods (Chaudhry et al., 2019). GSS (Aljundi et al., 2019c) extends the optimization perspective of GEM Lopez-Paz & Ranzato (2017), which necessitates prior knowledge of the number of tasks and task transitions, to an instance-based level and adds samples to the buffer based on their gradients. CoPE (De Lange & Tuytelaars, 2021) synchronizes the latent space with continually evolving prototypes with a high momentum-based update to facilitate learning. CN-DPM (Lee et al., 2020b) uses a Dirichlet process-based expansion mechanism aiming to increase the model's capacity, and SEDEM (Ye & Bors, 2023) improves the expansion process by introducing a self-assessment mechanism that considers the knowledge diversity among existing modules. Online-LoRA (Wei et al., 2025) uses loss dynamics to detect distribution shifts and fine-tunes pre-trained Vision Transformer models using a novel online weight regularization strategy to prevent forgetting. DCM (Ye & Bors, 2024) presents a novel memory management approach that dynamically adjusts memory clusters based on knowledge discrepancy criteria without task-specific signals. However, the practice of storing a subset of data for replay, expanding the network, or relying on large pre-trained architectures presents scalability challenges and is often constrained by factors such as privacy, memory limitations, and computational resources.

Network Pruning. Pruning removes some connections of the network and thereby limits the training process to a subset of parameters or neural paths. Mocanu et al. (2016a) showed that pruning a restricted Boltzmann machine at initialization achieves comparable performance to its dense counterpart. Similarly, Liu et al. (2022a) demonstrated the effectiveness of randomly pruned backbones when they are sufficiently wide, while Gadhikar et al. (2023) proved that randomly pruned random networks possess ample expressive capacity. Uniform (Mariet & Sra, 2015; He et al., 2017; Gale et al., 2019), Erdős-Rényi (ER) (Mocanu et al., 2018), and Erdős-Rényi Kernel (ERK) (Evci et al., 2020) are primary practical studies that rely on predetermined sparsity ratios but work well in practice. Recent research proposes more advanced approaches to adjust layer-wise sparsity ratios before training rather than relying on predetermined sparsity ratios (Lee et al., 2019; Wang et al., 2020; Tanaka et al., 2020). Instead of relying on weight magnitudes, edge-popup (Ramanujan et al., 2020) assigns distinct scores to each weight to guide the pruning process. This pruning strategy is employed by an offline continual learning approach SupSup (Wortsman et al., 2020) to construct subnetworks. However, it necessitates multiple forward and backward passes to determine the appropriate subnetwork for each task, limiting its applicability to online settings that demand each sample to be processed only once.

3 Self-Regulated Neurogenesis (SERENA)

In this section, we first share the background of existing online data-incremental learning methods and then provide our research question that motivates us to propose SERENA. Following, we give preliminaries and the details of training and inference of our approach which is illustrated in Figure 1 and positioned its novelty in Table 1.

Table 1: Online data-incremental learning baselines based on key characteristics.

	Requires Replay	Expands Network	Selects Connections
GSS	✓	×	×
CoPE	✓	×	×
MIR	✓	×	×
CN-DPM	✓	✓	×
Dynamic-OCM	✓	✓	×
SEDEM	✓	✓	×
SERENA (ours)	×	×	✓

Background and Motivation. Various methods proposed in online data-incremental learning can be divided into two main branches: replay-based and architecture-based. Replay-based CL approaches store a subset of the previous examples to be able to adjust the features and decision boundaries for all classes. Architecture-based approaches, on the other hand, learn the features of each concept or session on an entirely new model. However, they (i) bring overhead costs in terms of both memory and computation, (ii) violate privacy concerns, and (iii) complicate data retrieval and training. This leads us to an essential question in online data-incremental learning:

How can we enable continual learning without relying on complex procedures like initializing entirely new models or storing replay data while still promoting adaptation to new concepts and minimizing catastrophic forgetting?

To address this question, we benefit from the ability of deep neural networks to efficiently maintain multiple subnetworks or neural paths (Frankle & Carbin, 2019; Liu et al., 2020), allowing them to learn multiple tasks in a single model. This dynamic allocation of network connections enables models to preserve knowledge while adapting to new information, a crucial requirement in continual learning. Notably, this mechanism mirrors the brain's selective activation strategy, where specific neuron regions respond to particular stimuli (Babadi & Sompolinsky, 2014). Such selective responses facilitate efficient knowledge retrieval, prevent interference and aligns with *'self-regulated neurogenesis'* (Kempermann et al., 1998) and Hebbian Theory's principle (Löwel & Singer, 1992) of *'Neurons that fire together, wire together,'* which suggests that frequently co-activated neurons strengthen their connections over time, reinforcing task-specific knowledge in certain neural paths or circuits (Quiroga, 2012). By leveraging these principles, we aim to construct models that retain previously acquired representations while seamlessly integrating new knowledge into a single model, thereby bridging the gap between artificial and biological learning systems.

Algorithm 1: Pseudocode for SERENA

```
Input: Data stream set $\mathcal{S}$ with Batches of $\mathcal{B}$
       Model $\theta$
       Learning rate $\eta$
       Model sparsity ratio $m$
       Accuracy window $w$
       Accuracy threshold $t$

Eq. 1 to create sparse expert $\theta_m \subset \theta$
for $(x_i, y_i) \in \mathcal{B}$ do
    Eq. 2 to train $\theta_m$ with $\eta$
    if accuracy of $\mathcal{B} \leq \mu_w \cdot t$ then
        freeze $\theta_m$
        Eq. 1 to create new $\theta_m \subset \theta$
    end if
end for
```

Preliminaries. SERENA operates under an online data-incremental learning scenario which involves the continuous updating of a deep neural network model, where the data samples (x_i, y_i) arrive gradually from a set of non-iid data streams $\mathcal{S} = \{s_1, s_2, \ldots, s_s\}$ with processing batches of size B. Data sample i is constituted by input feature $x_i \in \mathcal{X}$ and corresponding signal y_i in which the output space for classification is a discrete set of classes $\mathcal{Y} \leftarrow \mathcal{Y}_{i-1} \cup y_i$. A deep network is optimized to map input instances to the classifier space in a flowing data stream, denoted as $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$, where θ represents the network parameters. In online data-incremental learning, samples correspond to a series of tasks without any identifiers, in contrast to task- or class-incremental setup where explicit task cues are necessary.

$$1 - \frac{(n^{l-1} + n^l + w^l + h^l)}{n^{l-1} n^l w^l h^l} \tag{1}$$

Concept Cell Allocation. SERENA uses zero-cost random unstructured pruning to the kernels of convolution layers and the connections of classifier head for each stream, using Erdős-Rényi Kernel (ERK) (Evci et al., 2020) as it enables better parameter efficiency, preserves model capacity in key regions, while suitable for longer data streams or larger number of tasks as demonstrated in prior work (Yildirim et al., 2023). Inspired by random graph theory, ERK allocates non-zero weights proportionally across layers based on the number of parameters, input/output channels and kernel dimensions, ensuring denser connectivity in the smaller layers. This allocation avoids over-pruning in the bottlenecks and has been shown to consistently outperform uniform sparsity. Formally, to obtain concept cell or neural path θ_m with overall sparsity level of binary mask m, it scales the sparsity of layers proportionally as given in Eq. 1, where n^l represents the number of neurons or feature maps at layer l, and w^l and h^l denote the width and height of the l^{th} convolutional kernel, aiming to preserve the information flow.

Drift Detection. SERENA incorporates a drift detection mechanism to continuously monitor the accuracy curve, enabling it to detect changes in the underlying concept being learned, similar to (Gama et al., 2004; Bifet & Gavalda, 2007). It tracks the accuracy of each batch of data $\mathcal{B}$ and compares it to the running accuracy over a defined window w. If the accuracy of the current batch $\mathcal{B}$ drops by more than a predetermined threshold t relative to the mean of running accuracy, SERENA interprets this as a significant deviation or concept change. In response, it creates a new neural path θ_m within the same fixed backbone θ to adapt to the evolving concept.

Training Flow. SERENA learns a different neural path or concept cell θ_m for each data stream s_s by minimizing the function given in Eq. 2 in an online manner. Here, CE denotes the Cross-Entropy Loss, θ_m denotes the randomly initialized concept cell, and the optimal θ^*_m is obtained by solving this optimization problem. When a new stream $\mathcal{S}$ is identified by a drift detector, it freezes these connections of θ^*_m against forgetting and allocates a new concept cell θ_m to train on, which can share connections with other concept cells. This allows using the predefined main backbone θ more efficiently while enabling knowledge transfer between the concept cells. We share the details in Algorithm 1.

$$\theta^*_m = \arg\min_{\theta_m} \sum_{i=1} [CE(f_{\theta_m}(x_i), y_i)] \tag{2}$$

Inference Flow. In evolving data streams, relying on a single selected path for classification can become unreliable due to distribution shifts, especially in the absence of explicit task or stream identifiers. Inspired by neurological processes, SERENA addresses this challenge through a novel ensemble classification strategy that mirrors how the brain tends to rely more heavily on the most recent experiences when making decisions, a phenomenon known as *recency effect* (Murdock Jr, 1962) or *recency bias* (Costabile & Klein, 2005). Formally, each concept cell θ_m produces a logit vector $z_i = f_{\theta_m}(x) \in \mathbb{R}^{|\mathcal{Y}|}$. SERENA then computes the final logits $\tilde{z}$ as a weighted sum of these outputs, as shown in Eq. 3, in which a linearly increasing weighting scheme prioritizes more recent neural paths while still integrating past knowledge. By dynamically adjusting the contributions of each concept cell, SERENA stabilizes decision boundaries and maintains robust predictions even without explicit task or stream labels. This neuro-inspired mechanism allows SERENA to be particularly effective in scenarios where pinpointing the correct neural path is inherently challenging, as its ensemble approach offers a reliable solution for managing evolving data distributions.

$$\tilde{z} = \sum_{i=1}^{\mathcal{S}} \frac{i}{\mathcal{S}} z_i, \quad \hat{y} = \arg\max_{y \in \mathcal{Y}} \tilde{z}_y \tag{3}$$

4 Experiments

In this section, we describe the experimental setup and present experiments conducted on twelve different benchmarks to evaluate the incremental learning capabilities of SERENA in comparison with other state-of-the-art algorithms. Additionally, we share a run-time *vs.* accuracy analysis as well as a space projection of the samples to provide the robustness of the proposed method in more detail.

4.1 Experimental Setup

Datasets and Scenarios. In balanced scenarios, **Split-MNIST** and **Split-CIFAR10** have 5 streams, and each stream has 2 classes with 12000 samples for MNIST (LeCun et al., 2010) and 10000 samples for CIFAR10 (Krizhevsky et al., 2009) in training and 2000 samples in testing.**Split-CIFAR100** and **Split-MiniImageNet** divide CIFAR100 (Krizhevsky et al., 2009) and MiniImageNet (Vinyals et al., 2016) into 20 streams, each with 5 disjoint classes with 2500 samples for training and 500 samples for testing. **Split-TinyImageNet** divides TinyImageNet into 100 streams, each consist of 2 classes with 1000 samples for training and 100 samples for testing. Finally, **Split-ImageNet** (Deng et al., 2009) has 20 streams, each with 50 classes with around 60000 samples for training and 2500 samples for testing. In imbalanced scenarios, sample sizes exhibit a logarithmic increase or decrease pattern. **Ascending Split-MNIST** gradually rises from 2500 to 12000 samples per stream. **Ascending Split-CIFAR10** gradually increases from 2000 to 10000 samples per stream. **Ascending Split-CIFAR100** starts from 500 up to 2500 samples per stream. The descending counterparts; namely **Descending Split-MNIST**, **Descending Split-CIFAR10**, and **Descending Split-CIFAR100** mirror this trajectory in reverse.

Compared Baselines. We compare with **finetune** as a naive approach; **reservoir** (Vitter, 1985), **GSS** (Aljundi et al., 2019c), **MIR** (Aljundi et al., 2019a), **CoPE** (De Lange & Tuytelaars, 2021) as replay-based approaches; and **CN-DPM** (Lee et al., 2020b), **Dynamic-OCM** (Ye & Bors, 2022), and **SEDEM** (Ye & Bors, 2023) as architecture expansion methods. Moreover, similar to (De Lange & Tuytelaars, 2021; Lee et al., 2020b), we compare with **iid-online** and **iid-offline** as upper reference points for performance by relaxing the non-iid property of CL to a traditional supervised learning setting where *online* trains all classes at once for a single epoch and *offline* for 50 epochs. Note that TIL and CIL methods are not designed or suitable for this setup; thus, they are not included as direct baselines for comparison.

Backbones. For Split-MNIST, we use a 2-layer MLP with 400 neurons in each layer before the classifier. We use standard a ResNet18 (He et al., 2016) for the rest. After each stream, we freeze used connections except batch normalization since they cannot be designated as stream-specific components.

Evaluation Metrics. We measure model performance with common CL metrics which are Average Accuracy and Forgetting (Shim et al., 2021; Gu et al., 2022). Average Accuracy is computed by training the model on the data stream comprising all streams and subsequently evaluating the final model's performance using the test data associated with all streams. It is formally defined as: $\frac{1}{S}\sum_{j=1}^{S} a_{S,j}$ where $a_{i,j}$ is the accuracy on stream j after the model is trained from stream 1 to i. Average Forgetting quantifies the extent to which the model forgets previously learned streams after being trained on the final stream and can be defined as: $\frac{1}{S-1}\sum_{j=1}^{S-1} f_{S,j}$ where $f_{i,j} = \max_{k\in(1,2,\ldots,i-1)} a_{k,j} - a_{i,j}$.

Implementation Details. We train from scratch with a batch size of 10, following (Shim et al., 2021; Guo et al., 2022; Wei et al., 2023). We set the window size to 10 and the accuracy drop threshold to 50% for the drift detector. We use SGD with a learning rate of 1×10^{-2} for MLP and 5×10^{-4} for ResNet18 and weight decay of 5×10^{-4} for both. When training new concept cells, we activate a fixed percentage of the weights, 20% for MLP and 5% for ResNet18. For the baselines, we use the same backbone and their default settings for a fair comparison. For online-joint, we train all classes at once for 1 epoch with a batch size of 10 and for offline-joint, 50 epochs with a batch size of 128.

4.2 Results and Analysis

Performance on balanced scenarios. In Table 2, the results demonstrate the superior performance of SERENA relative to state-of-the-art methods, highlighting its ability to learn continually. Furthermore, as illustrated in Figure 2, SERENA consistently achieves higher accuracy at every time step across all benchmarks, even in the absence of replay data (*M=0*) and network expansion. A key factor behind SERENA's success is its biologically inspired learning mechanism, which balances *plasticity* and *stability*. Existing approaches are generally good at plasticity by acquiring the knowledge of a given task while they suffer on the stability side by forgetting previously learned concepts, which can be observed in Table 4 and Figure 3. Unlike existing state-of-the-art methods, which struggle to maintain past knowledge even with the existence of replay and architecture expansion, SERENA achieves this naturally, ensuring strong retention of prior concepts without compromising adaptation to new streams.

Table 2: **Average accuracy [%]** (higher is better) of SERENA compared to existing methods on different benchmarks. The results of architecture expansion approaches are cited from Ye & Bors (2023) and denoted with †. Note that iid-offline and iid-online are not CL methods, and the best results are highlighted in bold.

Method	Split-MNIST *(M=2k)*	Split-CIFAR10 *(M=1k)*	Split-CIFAR100 *(M=5k)*	Split-MiniImageNet *(M=5k)*
iid-offline	98.4 ± 0.1	85.2 ± 1.1	58.3 ± 0.6	54.7 ± 0.9
iid-online	96.4 ± 0.2	59.9 ± 3.1	22.3 ± 3.2	19.2 ± 4.8
finetune	19.8 ± 0.1	17.6 ± 0.8	3.2 ± 0.3	3.7 ± 0.9
reservoir	92.2 ± 0.7	42.7 ± 2.7	20.7 ± 2.8	7.0 ± 2.3
GSS	91.9 ± 1.8	29.2 ± 4.2	14.5 ± 2.9	6.9 ± 3.3
CoPE	93.8 ± 0.6	50.8 ± 2.1	22.8 ± 1.5	13.4 ± 1.1
MIR	91.8 ± 3.5	39.3 ± 5.2	17.1 ± 1.7	7.4 ± 2.3
CN-DPM †	93.2 ± 0.1	45.2 ± 0.2	20.1 ± 0.1	27.9 ± 2.3
Dynamic-OCM †	94.0 ± 0.2	49.2 ± 1.5	21.8 ± 0.7	26.6 ± 2.1
SEDEM †	98.4 ± 0.2	55.3 ± 1.3	24.9 ± 1.2	29.6 ± 1.9
SERENA *(M=0)*	**98.5 ± 0.1**	**88.7 ± 0.9**	**62.1 ± 1.7**	**48.7 ± 1.8**

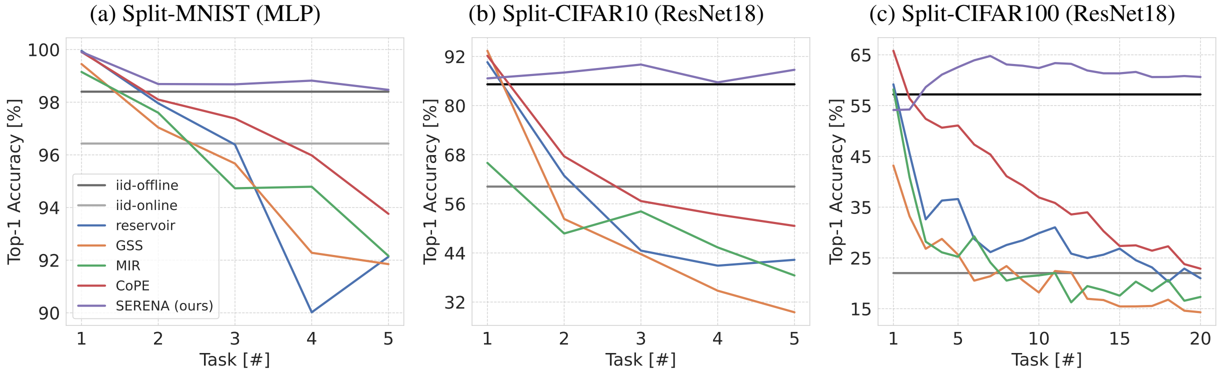


Figure 2: **Average accuracy [%]** after each stream on balanced benchmarks. SERENA outperforms all methods by a large margin, including iid-offline which is not a CL method but a standard supervised batch learning, representing an upper reference point.

Another remarkable advantage of SERENA is its ability to achieve performance levels comparable to iid-offline learning, making it particularly valuable for real-world applications where data is encountered sequentially. This capability bridges the gap between online continual learning and traditional offline training, demonstrating its potential for large-scale deployment. Notably, SERENA achieves this without initializing entirely new models or storing replay data, which is crucial for reducing computational costs, preserving privacy, and minimizing additional memory overhead.

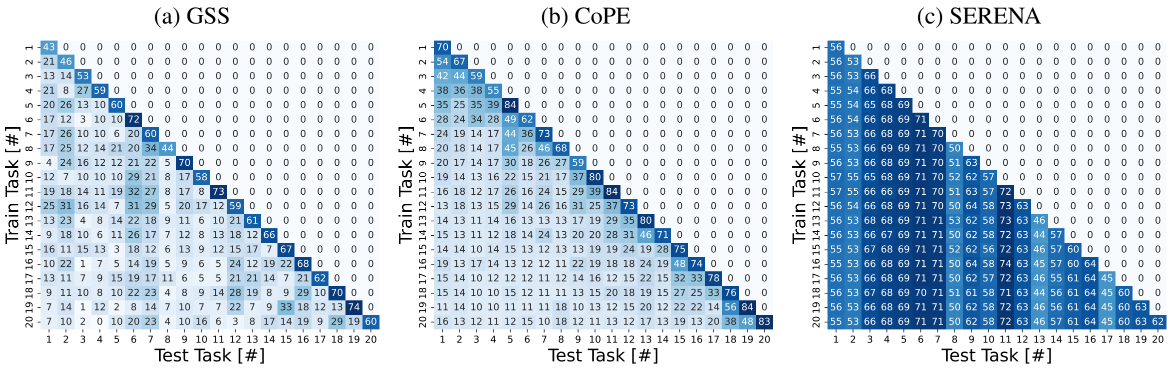
Table 3: **Average accuracy [%]** and **run time [hour]** on Split-ImageNet. SERENA reaches offline performance in an online data-incremental setup.

Method	Setup	Split-ImageNet	Run Time
iCaRL *(M=20k)*	*online*	2.11	6.6 hrs
	offline	14.96	64.2 hrs
FOSTER *(M=20k)*	*online*	0.94	11.4 hrs
	offline	37.76	80.7 hrs
SERENA *(M=0)*	*online*	36.22	4.1 hrs

Performance on a challenging scenario. We evaluated SERENA also on the large-scale ImageNet dataset, as shown in Table 3, by dividing it into 20 sequential streams, a setup we refer to as Split-ImageNet. This benchmark serves as a realistic and challenging test for continual learning, as it requires models to learn from high-dimensional, diverse visual data without revisiting previous concepts. Due to the absence of direct baselines in existing codebases, we were unable to compare SERENA directly with them. Instead, we benchmark against iCaRL Rebuffi et al. (2017) and FOSTER Wang et al. (2022a), two widely recognized methods designed for offline class-incremental learning that provide implementations compatible with ImageNet. To ensure a fair comparison, we evaluated both methods using a replay buffer of 20 images per class (totaling 20,000 images) in two different scenarios: *1-epoch online* and *50-epoch offline*. Our results reveal that iCaRL (Rebuffi et al., 2017) and FOSTER (Wang et al., 2022a) struggle significantly in the online setup, which aligns with expectations, as both methods are inherently designed for offline continual learning, where multiple training epochs help mitigate forgetting. While SERENA is specifically designed for the online setting, it substantially outperforms a fundamental method in offline continual learning iCaRL (Rebuffi et al., 2017) and achieves performance on par with the offline state-of-the-art method FOSTER (Wang et al., 2022a), despite operating in a far more challenging single-pass learning scenario. These results underscore SERENA's capacity to handle large-scale, real-world data streams efficiently without reliance on extensive memory buffers or multiple training iterations.

Table 4: **Average forgetting [%]** (lower is better) of SERENA compared to existing methods on different benchmarks. The results of architecture expansion approaches are cited from Ye & Bors (2023) and denoted with †, and "–" indicates that the result was not reported. Note that iid-offline and iid-online return zero forgetting since they are supervised learners, not continual learners.

Method	Split MNIST *(M=2k)*	Split CIFAR10 *(M=1k)*	Split CIFAR100 *(M=5k)*	Split-MiniImageNet *(M=5k)*
iid-offline	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
iid-online	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
finetune	79.5 ± 0.1	63.5 ± 2.2	58.6 ± 1.3	65.2 ± 2.1
reservoir	6.4 ± 0.1	41.9 ± 2.1	44.2 ± 2.4	48.9 ± 2.2
GSS	6.9 ± 1.7	58.4 ± 4.9	45.3 ± 5.1	47.2 ± 4.7
CoPE	4.3 ± 0.1	27.9 ± 2.8	36.6 ± 2.2	39.4 ± 2.4
MIR	6.8 ± 3.5	37.6 ± 9.8	39.9 ± 1.1	42.5 ± 1.5
CN-DPM †	–	–	–	–
Dynamic-OCM †	–	–	–	–
SEDEM †	–	–	–	–
SERENA *(M=0)*	**0.0 ± 0.0**	**0.7 ± 0.4**	**1.5 ± 1.1**	**1.7 ± 1.3**


(a) GSS

(b) CoPE

(c) SERENA

Figure 3: **Accuracy [%]** of each stream after the model has been trained on a new one for Split-CIFAR100. SERENA demonstrates its strength by maintaining consistently high accuracy and near-zero forgetting across the streams compared to existing approaches.

Table 5: Used connections in ResNet18 indicate that a significant portion of connections remains inactive, suggesting ample capacity to accommodate additional data streams.

Dataset	Activated Parameters [%]
Split-CIFAR10	18 ± 2
Split-CIFAR100	51 ± 2
Split-MiniImageNet	52 ± 3

Network Saturation. Random connection sharing among concept cells enables efficient network usage over a broader range of tasks. Table 5 illustrates that many connections stay inactive after processing all streams. For example, Split-CIFAR10 consists of 5 streams, and 18% of the network connections are utilized, despite each concept cell utilizing 5% of the connections. Similarly, Split-CIFAR100 has 20 streams, and only 51% of the connections are activated, showing that network usage does not increase linearly with the streams.

Performance on imbalanced scenarios. All methods experience a decline in average accuracy when exposed to the imbalanced data streams we introduce, with performance degradation becoming more pronounced as the degree of imbalance increases. However, SERENA demonstrates a significantly lower performance drop compared to other methods and consistently outperforms them, particularly in scenarios where the number of samples per stream is highly limited, as in Figure 4b and Figure 4c.

This improvement can be attributed to SERENA's novel mechanism of dynamically allocating new concept cells to incoming data streams. Unlike existing methods that struggle with class imbalance due to overemphasis on majority classes, SERENA effectively mitigates this bias by learning distinct representations on different neural paths for new data while preserving previously acquired knowledge. This ensures that rare or underrepresented classes receive sufficient attention, allowing for more balanced learning across different distributions.

By maintaining equal focus across all classes and streams, SERENA not only reduces the detrimental effects of imbalance but also achieves performance levels comparable to or exceeding those of 50-epoch offline supervised batch learning. This robustness is particularly critical in real-world applications where data is often collected in an online fashion and naturally exhibits class imbalances, requiring models that can handle continuous long-term performance and imbalanced learning.

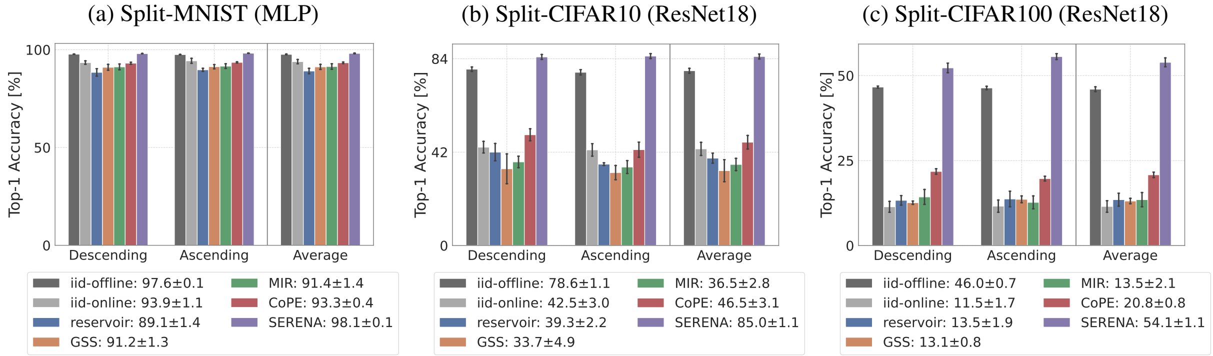


Figure 4: **Average accuracy [%]** on different imbalanced benchmarks. Although all methods experience some level of performance degradation, SERENA improves existing ones by a large margin, including the upper reference point iid-offline, which is not a CL method but supervised batch learning.

Run Time. SERENA maintains an on-par overall runtime to state-of-the-art methods while achieving significantly higher accuracy by utilizing only a small, randomly selected subnetwork for each stream. This selective activation reduces computational overhead during training and enhances efficiency in handling sequential learning tasks. For instance, on a single A100 GPU, SERENA completes Split-CIFAR10, Split-CIFAR100, and Split-MiniImageNet in approximately 5 minutes, 11 minutes, and 16 minutes respectively, matching the runtime of existing methods while nearly doubling the achieved accuracy, as illustrated in Figure 5. This efficiency is particularly crucial in online learning environments where rapid adaptation to new data is essential. By optimizing resource utilization without sacrificing performance, SERENA presents a practical solution for real-time continual learning scenarios, making it well-suited for large-scale applications with strict computational constraints.

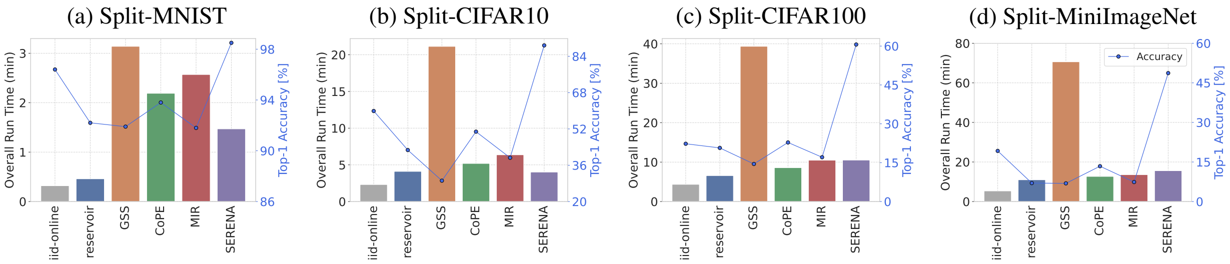


Figure 5: Overall runtime *vs.* average accuracy for each method. SERENA maintains a comparable runtime to existing methods while achieving significantly higher accuracy, demonstrating its efficiency in online data-incremental learning scenarios.

Ablation Study. We present the results in Table 6, which provides a comprehensive analysis of the performance of our models under different configurations, specifically varying in terms of sparsity levels, window sizes, drift detection thresholds, and learning rates, all evaluated on the Split-CIFAR10 benchmark. To better understand the sensitivity and robustness of our approach, we experimented with sparsity levels of 0.80, 0.95, and 0.99, each tested across multiple window sizes of 5, 10, and 20, and paired with drift detection thresholds of 0.1, 0.5, and 0.9. Additionally, we varied the learning rate across a range of values, including 0.0005, 0.001, 0.005, and 0.01, to observe the interaction between learning dynamics and structural compression.

From these experiments, we observe that extreme sparsity levels, particularly 0.99, or overly aggressive learning rates such as 0.01 frequently lead to a noticeable drop in final model accuracy. This suggests that excessive pruning or overly rapid updates result in unstable learning dynamics and hinder the model convergence to effectively capture meaningful patterns from the input data. On the other hand, a moderate sparsity with smaller updates tends to strike a favorable balance between model compactness and task performance.

Among all tested configurations, the best performance, reaching an accuracy of 89.7%, was achieved using a sparsity level of 0.95, a window size of 10, a drift threshold of 0.5, and a conservative learning rate of 0.0005. We believe this particular combination provides a sweet spot where the model can adapt adequately to distributional shifts between tasks without overreacting to noise or losing generalization.

Table 6: Ablation of different network sparsities, learning rates, window sizes, and threshold values on Split-CIFAR10 in terms of **average accuracy [%]**. Best result is highlighted in bold.

Sparsity Level		0.80			0.95			0.99		
Window Size	Drift Threshold	lr = 0.0005	lr = 0.001	lr = 0.01	lr = 0.0005	lr = 0.001	lr = 0.01	lr = 0.0005	lr = 0.001	lr = 0.01
5	0.1	89.0	89.3	41.0	89.5	88.3	40.9	88.8	89.3	41.1
	0.5	89.5	88.8	41.1	89.5	88.5	41.5	88.9	89.1	40.9
	0.9	89.1	88.7	41.1	87.8	88.3	41.3	89.5	88.7	40.6
10	0.1	89.3	88.9	41.1	89.5	88.3	40.9	89.5	89.2	40.8
	0.5	89.4	89.4	40.1	**89.7**	89.0	41.1	89.2	88.7	41.1
	0.9	88.8	88.8	41.0	89.6	87.6	39.9	89.5	89.1	41.2
20	0.1	89.2	89.2	41.2	89.5	88.5	41.0	89.3	88.9	41.1
	0.5	89.2	89.2	41.3	89.4	88.3	40.5	89.2	89.1	40.9
	0.9	80.3	89.1	40.9	80.0	88.6	41.1	89.4	89.2	40.9

5 Discussion

In this section, we first investigate how SERENA achieves performance comparable to offline learning by analyzing the representation space of the neural network. Furthermore, we examine two critical discussion points for continual learning: never-ending learning and class-revisiting scenarios. These settings pose unique challenges for fixed-capacity replay-free systems and demand novel strategies for efficient knowledge acquirement, retention, and reuse. Although not the central focus of our current experimental evaluation, we discuss how SERENA's working principles offer a strong foundation for addressing these complex scenarios.

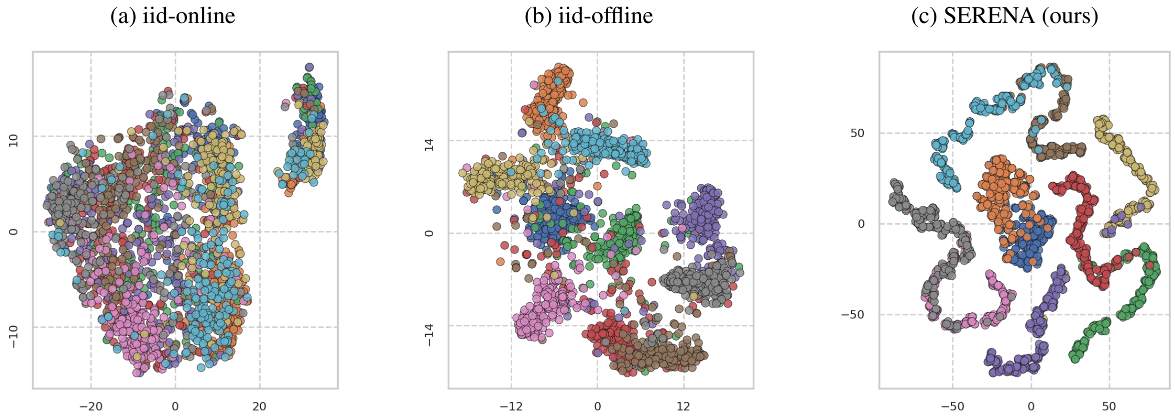


Figure 6: By learning specialized neural paths or concept cells Quiroga (2012) in a single pass, SERENA effectively separates distinct data streams and classes–as indicated by different colors–with minimal overlap.

Space Projection. To better understand why SERENA surpasses the upper reference point iid-offline, which uses 50-epoch supervised batch learning, we analyze the t-SNE projection of feature representations, *after training on all streams*. This analysis provides insight into how well different ways of training separate the streams and the classes within the same streams. As shown in Figure 6, the iid-online approach struggles to maintain distinct class boundaries, resulting in a near-random spread of samples, indicating poor feature separation. While the iid-offline approach improves upon this, showing clearer class differentiation, overlaps and moderate spread persist, increasing the risk of misclassification.

In contrast, SERENA exhibits well-defined class groups, achieving superior separation of each stream. Any observed overlap is negligible and typically occurs only between semantically related classes within the same stream. The curved structures in the t-SNE plot indicate that each network path or concept cell has learned a distinct manifold on top of the initial stream, which forms a simpler and more compact cluster (orange-blue pair). This remarkable ability highlights SERENA's effectiveness in structuring learned representations, making it highly resilient to forgetting. By dynamically leveraging distinct concept cells for each detected stream, SERENA ensures strong feature discrimination, facilitating continual learning while achieving the performance level of the offline supervised batch learning.

Table 7: **Average accuracy [%]** on Split-TinyImageNet, which simulates a never-ending scenario with 100 streams.

	Split-TinyImageNet	Used Parameters
	(M=4k)	(from 11M)
finetune	1.7	11M (100%)
reservoir	2.0	11M (100%)
GSS	2.4	11M (100%)
CoPE	3.2	11M (100%)
MIR	2.9	11M (100%)
SERENA (M=0)	**56.9**	9.8M (89%)

Never-Ending Scenario. Network saturation and reduced plasticity are inherent limitations of many fixed-resource replay-free continual learning methods, including SERENA. Nevertheless, findings suggest that our approach offers a promising direction for such scenarios. Its design allows for efficient parameter utilization by reusing existing connections across streams rather than allocating entirely new parameters for each stream and this significantly delays the saturation without compromising performance. Our empirical analysis in Table 5 substantiates this claim by demonstrating a sublinear growth in parameter usage by overlap across different neural paths. To emulate a never-ending scenario, we designed an experimental setup involving 100 sequential streams using TinyImageNet (Le & Yang, 2015) in Table 7. Notably, SERENA surpasses existing approaches both in performance and efficiency. While infinite stream lengths will inevitably exhaust any fixed model capacity, SERENA's ability to sustain learning across long task sequences with minimal redundancy represents a significant step toward scalable, never-ending scenarios.

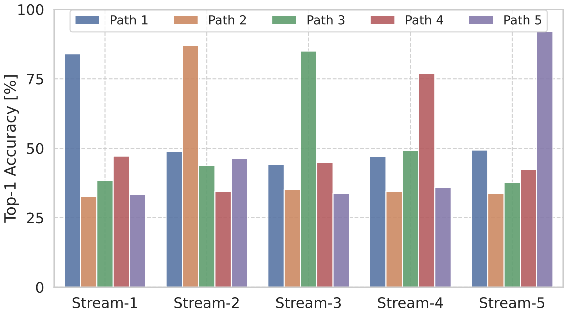


Figure 7: Neural path performances on each stream, after completing all learning sessions on Split-CIFAR10.

Class-Revisiting Scenario. We conduct extensive evaluations of SERENA on widely recognized online continual learning benchmarks to ensure fair comparisons. While those benchmarks do not explicitly consider repetitive stream scenarios, we acknowledge that such settings represent a compelling and realistic extension of the continual learning. Despite their practical relevance, task revisiting scenarios are not yet standardized benchmarks in the field and thus remain mostly underexplored. Therefore, it is an important challenge that reflects a broader and largely unexplored area in continual learning. For example, replay-based methods store samples from all the tasks they encounter without discerning whether a task has truly changed or repeated. Similarly, architecture expansion methods typically allocate new parameters for each incoming task without verifying task novelty, resulting in unnecessary model growth. Although SERENA does not currently implement a dedicated mechanism for task recurrence, its architectural design offers natural extensibility to such cases. Specifically, SERENA's integration of concept drift detection and stream-specific neural paths enables the evaluation of existing paths upon encountering a new input stream. If a previously established path yields high predictive accuracy, as proved in Figure 7, it could be reused and interpreted as a sign of task recurrence rather than allocating additional resources. This potential for reuse not only mitigates unnecessary parameter growth but also highlights SERENA's flexibility and efficiency, making it a strong candidate for future research in class-revisiting continual learning.

6 Conclusion

Existing online data-incremental learning approaches rely heavily on complex mechanisms, such as storing replay data or expanding the model with entirely new network architectures, both of which introduce significant overhead costs and potential privacy concerns. Inspired by self-regulated neurogenesis (Kempermann et al., 1998; Yiu & He, 2006) and concept cells (Quiroga, 2012) observed in the human brain, we introduce SERENA for online data-incremental learning. It dynamically adapts to concept drift by continuously fine-tuning the model and allocating neural connections or pathways for new concepts without storing past data or expanding the initial architecture. This enables SERENA to learn efficiently in a streaming setting while maintaining stability and minimizing interference. Additionally, we introduce new continual learning scenarios designed to better reflect real-world data dynamics where the sample sizes gradually increase or decrease across learning sessions. Our extensive experiments demonstrate that SERENA not only surpasses state-of-the-art methods but also surpasses offline supervised batch learning performance. This underscores its effectiveness in real-world applications where storage constraints, computational efficiency, and privacy considerations are critical. By eliminating the need for replay data and avoiding unnecessary architectural expansion, SERENA provides a scalable and biologically inspired solution for continual learning.

Limitations and Future Work. While SERENA efficiently utilizes network capacity, prolonged learning across an increasing number of streams may lead to model saturation, potentially limiting its ability to accommodate new tasks. Future work could explore adaptive mechanisms for optimizing network capacity and other continual learning scenarios, such as blurry and few-shot setups.

Broader Impact

This paper presents work whose goal is to advance the field of Machine Learning, especially on the subject of Online Data-Incremental Learning. Besides the advancements in the field, it eliminates the need to store data or expand models, thereby diminishing privacy, memory, computation, and scalability concerns.

Acknowledgements

This work is supported by SYNERGIES, a project funded by the EU Horizon programme under GA No. 101146542; Dutch national e-infrastructure with the support of SURF Cooperative using GA no. EINF-10242; and Turkish MoNE scholarship.

References

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, 2017.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018.

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, et al. Online continual learning with maximal interfered retrieval. In *NeurIPS*, 2019a.

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *CVPR*, 2019b.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, 2019c.

Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *ICLR*, 2022.

Duncan E Astle and Gaia Scerif. Interactions between attention and visual short-term memory (vstm): What can be learnt from individual and developmental differences? *Neuropsychologia*, 49(6):1435–1445, 2011.

Baktash Babadi and Haim Sompolinsky. Sparseness and expansion in sensory representations. *Neuron*, 83(5), 2014.

Albert Bifet and Ricard Gavalda. Learning from time-changing data with adaptive windowing. In *International Conference on Data Mining - SIAM*. SIAM, 2007.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 2018.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, P Dokania, P Torr, and M Ranzato. Continual learning with tiny episodic memories. In *Workshop on Multi-Task and Lifelong Reinforcement Learning*, 2019.

Kristi A Costabile and Stanley B Klein. Finishing strong: Recency effects in juror judgments. *Basic and Applied Social Psychology*, 2005.

Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *ICCV*, 2021.

Aleksandr Dekhovich, David MJ Tax, Marcel HF Sluiter, and Miguel A Bessa. Continual prune-and-select: class-incremental learning with specialized subnetworks. *Applied Intelligence*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *CVPR*, 2022.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *ICML*, 2020.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*, 2019.

Advait Harshal Gadhikar, Sohom Mukherjee, and Rebekka Burkholz. Why random pruning is all we need to start sparse. In *ICML*, 2023.

Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv:1902.09574*, 2019.

Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In *"Advances in Artificial Intelligence - SBIA"*, 2004.

Siavash Golkar, Michael Kagan, and Kyunghyun Cho. Continual learning via neural pruning. In *NeurIPS*, 2019.

Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 478(2266), 2022.

Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng. Not just selection, but exploration: Online class-incremental continual learning via dual view consistency. In *CVPR*, 2022.

Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *ICML*, 2022.

Mustafa Burak Gurbuz and Constantine Dovrolis. Nispa: Neuro-inspired stability-plasticity adaptation for continual learning in sparse networks. In *ICML*, 2022.

Chen He, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exemplar-supported generative reproduction for class incremental learning. In *BMVC*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *ICCV*, 2017.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Lifelong learning via progressive distillation and retrospection. In *ECCV*, 2018.

Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao Tao, Dongyan Zhao, Jinwen Ma, and Rui Yan. Overcoming catastrophic forgetting for continual learning via model adaptation. In *ICLR*, 2019.

Yusong Hu, De Cheng, Dingwen Zhang, Nannan Wang, Tongliang Liu, and Xinbo Gao. Task-aware orthogonal sparse network for exploring shared knowledge in continual learning. In *ICML*, 2024.

Haeyong Kang, Rusty John Lloyd Mina, et al. Forget-free continual learning with winning subnetworks. In *ICML*, 2022a.

Haeyong Kang, Jaehong Yoon, Sultan Rizky Hikmawan Madjid, Sung Ju Hwang, and Chang D. Yoo. On the soft-subnetwork for few-shot class incremental learning. In *ICLR*, 2023.

Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *CVPR*, 2022b.

Gerd Kempermann, H Georg Kuhn, and Fred H Gage. Experience-induced neurogenesis in the senescent dentate gyrus. *Journal of Neuroscience*, 18(9), 1998.

James Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. 2015.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs*, 2, 2010.

Janghyeon Lee, Hyeong Gwon Hong, Donggyu Joo, and Junmo Kim. Continual learning with extended kronecker-factored approximate curvature. In *CVPR*, 2020a.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *ICLR*, 2019.

Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. In *ICLR*, 2020b.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 2017.

Shiwei Liu, Tim Van der Lee, Anil Yaman, Zahra Atashgahi, Davide Ferraro, Ghada Sokar, Mykola Pechenizkiy, and Decebal Constantin Mocanu. Topological insights into sparse neural networks. In *ECML PKDD*, 2020.

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Li Shen, Decebal Constantin Mocanu, Zhangyang Wang, and Mykola Pechenizkiy. The unreasonable effectiveness of random pruning: Return of the most naive baseline for sparse training. In *ICLR*, 2022a.

Xialei Liu, Yu-Song Hu, Xu-Sheng Cao, Andrew D Bagdanov, Ke Li, and Ming-Ming Cheng. Long-tailed class incremental learning. In *ECCV*, 2022b.

Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *CVPR*, 2021.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, 2017.

Frederico Lourenco and BJ Casey. Adjusting behavior to changing environmental demands with development. *Neuroscience & Biobehavioral Reviews*, 37(9):2233–2242, 2013.

Siegrid Löwel and Wolf Singer. Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity. *Science*, 255(5041):209–212, 1992.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, 2018.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, 2018.

Zelda Mariet and Suvrit Sra. Diversity networks: Neural network compression using determinantal point processes. *arXiv:1511.05077*, 2015.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Elsevier, 1989.

Decebal Constantin Mocanu, Elena Mocanu, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. A topological insight into restricted boltzmann machines. *Machine Learning*, 104:243–270, 2016a.

Decebal Constantin Mocanu, Maria Torres Vega, et al. Online contrastive divergence with generative replay: Experience replay without storing data. *arXiv:1610.05555*, 2016b.

Decebal Constantin Mocanu et al. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 2018.

Bennet B Murdock Jr. The serial position effect of free recall. *Journal of experimental psychology*, 1962.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *WACV*, 2023.

Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. In *NeurIPS*, 2021.

Rodrigo Quian Quiroga. Concept cells: the building blocks of declarative memory functions. *Nature Reviews Neuroscience*, 13(8), 2012.

Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *CVPR*, 2020.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017.

Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Error sensitivity modulation based experience replay: Mitigating abrupt representation drift in continual learning. In *ICLR*, 2023.

Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. In *AAAI*, 2021.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NeurIPS*, 2017.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, et al. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, 2023.

Ghada Sokar, Decebal Constantin Mocanu, and Mykola Pechenizkiy. Spacenet: Make free space for continual learning. *Neurocomputing*, 2021.

Ghada Sokar, Decebal Constantin Mocanu, and Mykola Pechenizkiy. Avoiding forgetting and allowing forward transfer in continual learning via sparse networks. In *ECML PKDD*, 2023.

Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *NeurIPS*, 2020.

Ben Tsuda, Kay M Tye, Hava T Siegelmann, and Terrence J Sejnowski. A modeling framework for adaptive lifelong learning with transfer and savings through gating in the prefrontal cortex. *Proceedings of the National Academy of Sciences*, 117(47), 2020.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, 2016.

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

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *ICLR*, 2020.

Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *ECCV*, 2022a.

Wenjin Wang, Yunqing Hu, Qianglong Chen, and Yin Zhang. Task difficulty aware parameter allocation & regularization for lifelong learning. In *CVPR*, 2023.

Zifeng Wang, Zheng Zhan, Yifan Gong, Geng Yuan, Wei Niu, Tong Jian, Bin Ren, Stratis Ioannidis, Yanzhi Wang, and Jennifer Dy. Sparcl: Sparse continual learning on the edge. In *NeurIPS*, 2022b.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, 2022c.

Xiwen Wei, Guihong Li, and Radu Marculescu. Online-lora: Task-free online continual learning via low rank adaptation. In *WACV*, 2025.

Yujie Wei, Jiaxin Ye, Zhizhong Huang, Junping Zhang, and Hongming Shan. Online prototype learning for online continual learning. In *ICCV*, 2023.

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. In *NeurIPS*, 2020.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, 2019.

Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *CVPR*, 2021.

Fei Ye and Adrian G Bors. Continual variational autoencoder learning via online cooperative memorization. In *ECCV*, 2022.

Fei Ye and Adrian G. Bors. Self-evolved dynamic expansion model for task-free continual learning. In *ICCV*, 2023.

Fei Ye and Adrian G Bors. Online task-free continual generative and discriminative learning via dynamic cluster memory. In *CVPR*, 2024.

Murat Onur Yildirim, Elif Ceren Gok Yildirim, Ghada Sokar, Decebal Constantin Mocanu, and Joaquin Vanschoren. Continual learning with dynamic sparse training: Exploring algorithms for effective model updates. In *CPAL*, 2023.

Murat Onur Yildirim, Elif Ceren Gok Yildirim, and Joaquin Vanschoren. Sculpting [cls] features for pre-trained model-based class-incremental learning. *arXiv:2502.14762*, 2025.

Glenn Yiu and Zhigang He. Glial inhibition of cns axon regeneration. *Nature Reviews Neuroscience*, 7(8), 2006.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017.

Chen Zeno, Itay Golan, Elad Hoffer, and Daniel Soudry. Task agnostic continual learning using online variational bayes. *arXiv:1902.09574*, 2019.

Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *CVPR*, 2020.

Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *ICLR*, 2023.

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *CVPR*, 2024.

Xinyun Zou, Soheil Kolouri, Praveen K Pilly, and Jeffrey L Krichmar. Neuromodulated attention and goal-driven perception in uncertain domains. *Neural Networks*, 125, 2020.