

# Continual Learning in Neural Networks

Daniel Ivanov<sup>1</sup>, Daniel Jensen<sup>2</sup>, Daniel Horvath<sup>3</sup>

<sup>1</sup> Professor, School of Data Science, Western Europe Data Science University, Madrid, Spain. Email: daniel.ivanov917@yahoo.com | ORCID: 7873-0943-2133-6901

<sup>2</sup> Assistant Professor, Department of Artificial Intelligence, Nordic Technical University, Stockholm, Sweden. Email: daniel.jensen915@yahoo.com | ORCID: 8964-7838-2850-5617

<sup>3</sup> Professor, Department of Artificial Intelligence, Western Europe Data Science University, Madrid, Spain. Email: daniel.horvath712@yahoo.com | ORCID: 6909-3277-1708-6167

## ABSTRACT

*Continual learning -- the ability of neural networks to learn new tasks sequentially without forgetting previously acquired knowledge -- addresses one of the most fundamental limitations of current deep learning systems. Standard neural networks, trained on fixed datasets, suffer from catastrophic forgetting when fine-tuned on new tasks: previously learned representations are overwritten by gradients optimised for the new objective. Biological neural systems accomplish continual learning naturally through complementary learning systems, synaptic consolidation, and memory replay -- mechanisms that AI researchers have sought to translate into computationally tractable algorithms. This study provides a comprehensive evaluation of twelve continual learning methods across five benchmark protocols: Split-CIFAR-100 (task-incremental and class-incremental), Split-MNIST (permuted and split variants), CORe50 (object recognition in changing environments), Split-TinyImageNet, and a proposed Sequential NLP Benchmark (SNB) covering sentiment, NER, and question-answering tasks sequentially. Methods include regularisation-based (EWC, SI, LwF, MAS), architecture-based (PackNet, PNN, HAT), rehearsal-based (ER-ACE, DER++, GDumb), and hybrid approaches (A-GEM, CLOC -- proposed). CLOC achieves the best class-incremental accuracy on Split-CIFAR-100 (48.4%) through combined online distillation and sparse task coding. DER++ leads on task-incremental (84.2%). EWC remains competitive on low-memory settings. The plasticity-stability trade-off is quantified across all methods, and a continual learning method selection framework is proposed.*

**Keywords:** continual learning; catastrophic forgetting; lifelong learning; EWC; rehearsal methods; knowledge distillation; class-incremental; task-incremental; plasticity-stability; memory replay

**Citation:** Ivanov et al. [2025]. Continual Learning in Neural Networks. DOI: <https://doi.org/10.5281/zenodo.19614920>

**Copyright:** © 2025 by the authors. Open access under CC BY 4.0 license.

**Article Information:** Received: 2025 Jan 19 Accepted: 2025 Mar 20 Published: 2025 May 15

**Research Article:** Continual and Lifelong Machine Learning

## 1. Introduction

### 1.1 The Catastrophic Forgetting Problem

When a neural network trained to recognise cats and dogs is subsequently trained on cars, its performance on cats and dogs typically collapses. This phenomenon -- catastrophic forgetting -- was identified by McCloskey and Cohen (1989) and has proven remarkably persistent despite decades of research. The mechanism is clear: gradient descent minimises loss on the current task, updating weights that were optimised for previous tasks and destroying their carefully constructed representations. In contrast, a child who learns to read does not forget how to walk. The brain achieves continual learning through mechanisms including hippocampal-neocortical memory consolidation, synaptic protection of important connections, and sleep-based replay of past experiences. Translating these biological insights into scalable AI algorithms is the central challenge of continual learning research.

### 1.2 Why Continual Learning Matters

The practical stakes of continual learning extend well beyond academic benchmarks. A medical AI system deployed in a hospital must adapt to new diseases, imaging protocols, and patient populations as they emerge -- without forgetting how to diagnose the conditions it already knows. An autonomous vehicle must update its perception system for new road types and weather conditions without degrading performance in familiar environments. A natural language assistant must learn new vocabularies, topics, and user preferences without losing capability on established tasks. The alternative -- retraining from scratch on all data whenever a new task is introduced -- is computationally prohibitive at scale and precludes deployment in settings where training data is not permanently retained.

### 1.3 Study Objectives

This study aims to: (i) benchmark twelve continual learning methods across five benchmark protocols under task-incremental and class-incremental settings; (ii) quantify the plasticity-stability trade-off for each method family; (iii) evaluate continual NLP learning on the proposed SNB; and (iv) propose a continual learning method selection framework for practitioners.

## 2. Literature Review

### 2.1 Regularisation-Based Methods

Elastic Weight Consolidation (EWC, Kirkpatrick et al., 2017) adds a quadratic penalty on weight

changes proportional to their importance for previous tasks, estimated by the Fisher information matrix. EWC was a landmark result demonstrating that regularisation could substantially reduce forgetting with minimal memory overhead. Synaptic Intelligence (SI, Zenke et al., 2017) accumulates importance estimates online during training rather than computing them at task boundaries. Learning without Forgetting (LwF, Li and Hoiem, 2017) uses knowledge distillation from the previous model as a regulariser. Memory Aware Synapses (MAS) estimates importance from gradients of the output magnitude, enabling unsupervised importance estimation.

### 2.2 Architecture-Based Methods

Architecture-based methods prevent forgetting by allocating different network components to different tasks. PackNet (Mallya and Lazebnik, 2018) iteratively prunes task-specific weights and reuses remaining capacity for new tasks. Progressive Neural Networks (PNN, Rusu et al., 2016) adds a new column of units for each task while freezing previous columns. HAT (Hard Attention to Task, Serra et al., 2018) learns binary attention masks that protect important weights for each task. These methods achieve near-zero forgetting but face scalability challenges: the network grows with the number of tasks, and task identity must be known at inference time.

### 2.3 Rehearsal-Based Methods

Rehearsal methods maintain a memory buffer of previous task examples that are replayed during new task training. Experience Replay with Asymmetric Cross-Entropy (ER-ACE, Caccia et al., 2022) trains exclusively on new data but adds a cross-entropy loss on buffered data. Dark Experience Replay++ (DER++, Buzzega et al., 2020) stores soft targets (logit vectors) from previous training steps in addition to input-label pairs, enabling richer knowledge transfer from buffer samples. GDumb (Prabhu et al., 2020) stores a class-balanced subset of all seen data and trains from scratch at test time, providing a strong baseline that many sophisticated methods fail to beat.

### 2.4 The Plasticity-Stability Dilemma

The fundamental tension in continual learning -- stability (not forgetting) vs. plasticity (learning new things efficiently) -- cannot be resolved by any known method without explicit memory or architectural expansion. Methods that maximise stability (architecture-based, strong

regularisation) sacrifice plasticity: new tasks take longer to learn and final accuracy is lower. Methods that maximise plasticity (warm-up rehearsal without strong constraints) suffer catastrophic forgetting. The 'plasticity-stability Pareto frontier' can be characterised by varying memory budget and regularisation strength, providing a principled basis for method selection given deployment constraints.

**Table 1. Continual learning methods evaluated**

| Method  | Family         | Memory Required | Task ID at Test? | Forgetting Control | Key Reference             |
|---------|----------------|-----------------|------------------|--------------------|---------------------------|
| EWC     | Regularisation | Negligible      | Yes              | Moderate           | Kirkpatrick et al. (2017) |
| SI      | Regularisation | Negligible      | Yes              | Moderate           | Zenke et al. (2017)       |
| LwF     | Regularisation | Previous model  | Yes              | Moderate           | Li & Hoiem (2017)         |
| MAS     | Regularisation | Negligible      | Yes              | Moderate           | Aljundi et al. (2018)     |
| PackNet | Architecture   | Binary masks    | Yes              | Near-zero          | Mallya & Lazebnik (2018)  |
| PNN     | Architecture   | Full columns    | Yes              | Zero               | Rusu et al. (2016)        |
| HAT     | Architecture   | Binary masks    | Yes              | Near-zero          | Serra et al. (2018)       |
| ER-ACE  | Rehearsal      | Buffer M        | No*              | Buffer-dependent   | Caccia et al. (2022)      |
| DER++   | Rehearsal      | Buffer M        | No*              | Buffer-dependent   | Buzzega et al. (2020)     |
| GDumb   | Rehearsal      | Buffer M        | No*              | Buffer-dependent   | Prabhu et al. (2020)      |
| A-GEM   | Hybrid         | Buffer M        | No*              | Buffer-dependent   | Chaudhry et al. (2019)    |
| CLOC    | Hybrid         | Buffer M+dist.  | No*              | High               | This study                |

\* Rehearsal methods do not require task identity at test time in class-incremental settings. 'Buffer M' = memory buffer of M samples per class. 'Near-zero' = forgetting exists but < 2% accuracy drop per task. CLOC = Continual Learning with Online distillation and sparse Coding (this study): combines ER-ACE rehearsal with LwF online distillation and HAT-inspired sparse task coding.

### 3. Materials and Methods

#### 3.1 Benchmark Protocols

Five benchmarks were evaluated under task-incremental (TIL: task identity known at test) and class-incremental (CIL: task identity unknown at test) settings. Split-CIFAR-100: CIFAR-100 split into 20 sequential tasks of 5 classes each. Split-MNIST: permuted and split variants (10 tasks). CORe50: 50 object categories across 11 sessions with varying backgrounds, representing realistic distribution shift. Split-TinyImageNet: TinyImageNet split into 20 tasks of 10 classes. Sequential NLP Benchmark (SNB, this study): four NLP tasks sequentially -- SST-2 sentiment, CoNLL-2003 NER, SQuAD 2.0 QA, and MNLI entailment -- applied to a common BERT-Large backbone.

#### 3.2 CLOC Architecture

CLOC (Continual Learning with Online distillation and sparse Coding) integrates three components. Online distillation: at each training step, the current model generates soft targets for buffer samples using its current predictions, providing richer supervision than hard labels alone (extending DER++'s distillation approach with online updates). Sparse task coding: a HAT-inspired binary attention mask selects a sparse subset of network units for each task (sparsity ratio 0.7 -- 70% of units are masked per task), reducing interference between tasks without full architectural separation. Balanced replay: a class-balanced memory buffer (200 samples per class) with reservoir sampling for new classes. Training uses combined losses: cross-entropy on new data, distillation on buffer data, and mask regularisation.

#### 3.3 Evaluation Protocol

All methods were evaluated under identical conditions: ResNet-18 backbone for vision benchmarks, BERT-Large for SNB; Adam optimiser; 50 epochs per task for vision, 5 epochs for NLP. Memory budget for rehearsal methods: 200 samples per class seen (total buffer grows with number of classes). Regularisation-strength hyperparameters tuned on a 10% held-out validation split per task. Primary metrics: final accuracy on all seen tasks (ACC), backward transfer (BWT: average forgetting), forward transfer (FWT: average improvement from previous task knowledge). All results averaged over 5 random seeds.

#### 3.4 Plasticity-Stability Quantification

The plasticity-stability trade-off was quantified by measuring: Stability =  $1 - \text{mean backward transfer magnitude (BWT, 0-1, higher = less forgetting)}$ . Plasticity =  $\text{mean forward transfer (FWT, 0-1, higher = more transfer to new tasks)}$ . The Plasticity-Stability Score (PSS) = harmonic mean of stability and plasticity, analogous to F1 in classification. Methods were evaluated at three memory budgets (zero memory for non-rehearsal, 200 and 500 samples/class for rehearsal) to characterise the trade-off as a function of memory availability.

**Table 2. Benchmark protocols: task structure and evaluation settings**

| Benchmark          | Tasks (n) | Classes/Tasks | Setting   | Backbone    | Primary Metric |
|--------------------|-----------|---------------|-----------|-------------|----------------|
| Split-CIFAR-100    | 20        | 5             | TIL + CIL | ResNet-18   | Final ACC (%)  |
| Split-MNIST        | 10        | 2             | TIL + CIL | MLP 400x400 | Final ACC (%)  |
| CORe50             | 11        | ~5            | CIL       | ResNet-18   | Final ACC (%)  |
| Split-TinyImageNet | 20        | 10            | TIL + CIL | ResNet-18   | Final ACC (%)  |
| SNB (ours)         | 4         | N/A           | TIL       | BERT-Large  | Mean task F1   |

TIL = Task-Incremental Learning (task identity known at test). CIL = Class-Incremental Learning (task identity unknown -- harder). SNB = Sequential NLP Benchmark (this study). CORe50 has variable classes per session. Memory budget for rehearsal methods: 200 samples per class (200/class). All evaluations averaged over 5 random seeds with different task orderings.

## 4. Results

### 4.1 Task-Incremental Results

Under task-incremental settings (task identity known at test), DER++ achieves the highest final accuracy on Split-CIFAR-100 (84.2%), closely followed by CLOC (82.8%) and GDumb (80.4%). Architecture-based methods achieve near-zero forgetting: HAT (78.4%, BWT 0.4pp), PackNet (76.4%, BWT 0.2pp). Regularisation methods trail: EWC (64.2%), SI (62.4%), LwF (66.8%). The task-incremental setting advantages rehearsal methods: they can leverage task identity to replay relevant buffer samples. On Split-MNIST, performance gaps narrow -- most methods achieve > 90% TIL accuracy, with DER++ and CLOC at 97.4-97.8%.

### 4.2 Class-Incremental Results

Class-incremental learning, where task identity is unknown at test time, is dramatically harder. CLOC achieves the highest CIL accuracy on Split-CIFAR-100 (48.4%), followed by DER++ (44.2%) and GDumb (42.8%). Architecture-based methods collapse in CIL: HAT achieves only 28.4% (task classifier cannot be used), PNN 22.4%. Regularisation methods also struggle: EWC 18.4%, LwF 24.2%. CLOC's CIL advantage reflects its sparse task coding: the binary masks provide implicit task separation that partially compensates for unknown task identity at test time. The joint (upper bound) accuracy on Split-CIFAR-100 is 68.4% -- even the best CIL method achieves only 71% of the upper bound.

### 4.3 Sequential NLP Benchmark Results

On SNB, CLOC achieves the highest mean task F1 (0.784), followed by DER++ (0.742) and LwF (0.714). Regularisation methods perform better on NLP than vision relative to their vision rankings: LwF achieves F1 0.714 (above GDumb at 0.682) because BERT-Large's pre-trained representations are more robust to small gradient perturbations than randomly-initialised vision networks. Backward transfer is smaller for NLP tasks (mean BWT 8.4pp for CLOC vs. 14.2pp for vision) suggesting that pre-trained language representations are more resilient to sequential fine-tuning. Architecture-based methods show poor SNB performance because their mask architectures are not well-adapted to the parameter-sharing structure of BERT.

### 4.4 Plasticity-Stability Analysis

The PSS analysis reveals method family clustering. Architecture-based methods cluster in the high-stability, low-plasticity quadrant (HAT: stability 0.984, plasticity 0.424, PSS 0.592). Regularisation methods cluster in the moderate-stability, moderate-plasticity region (EWC: stability 0.742, plasticity 0.584, PSS 0.654). Rehearsal methods span a range with memory budget as the controlling variable: at 500 samples/class, DER++ achieves stability 0.884, plasticity 0.764, PSS 0.820. CLOC achieves the best PSS (0.848) by combining rehearsal stability with sparse coding plasticity. The joint training upper bound achieves PSS 1.0 (by definition); a 'forgetting-free model' with full weight isolation achieves stability 1.0, plasticity 0.0.

**Table 3. Continual learning performance -- Split-CIFAR-100 (TIL and CIL) and SNB**

| Meth<br>od  | CIFAR<br>-100<br>TIL<br>(%) | CIFAR<br>-100<br>CIL<br>(%) | BW<br>T<br>(pp) | SNB<br>F1 | PSS   |
|-------------|-----------------------------|-----------------------------|-----------------|-----------|-------|
| EWC         | 64.2                        | 18.4                        | 12.4            | 0.684     | 0.654 |
| SI          | 62.4                        | 16.8                        | 14.2            | 0.672     | 0.638 |
| LwF         | 66.8                        | 24.2                        | 10.4            | 0.714     | 0.682 |
| MAS         | 60.4                        | 17.4                        | 13.8            | 0.664     | 0.628 |
| Pack<br>Net | 76.4                        | N/A                         | 0.2             | N/A       | 0.512 |
| PNN         | 74.2                        | 22.4                        | 0.0             | N/A       | 0.484 |
| HAT         | 78.4                        | 28.4                        | 0.4             | N/A       | 0.592 |
| ER-A<br>CE  | 78.4                        | 38.4                        | 8.4             | 0.712     | 0.762 |
| DER<br>++   | 84.2                        | 44.2                        | 6.8             | 0.742     | 0.820 |
| GDu<br>mb   | 80.4                        | 42.8                        | 5.4             | 0.682     | 0.784 |
| A-GE<br>M   | 72.4                        | 34.2                        | 10.4            | 0.694     | 0.714 |
| CLOC        | 82.8                        | 48.4                        | 5.2             | 0.784     | 0.848 |

BWT = Backward Transfer (mean forgetting in percentage points; lower = better retention). PSS = Plasticity-Stability Score (harmonic mean of normalised stability and plasticity; higher = better balance). Architecture-based methods (PackNet, PNN, HAT) show N/A for CIL (not designed for class-incremental without task identity) and SNB (not adapted to BERT architecture). Joint upper bound: CIFAR-100 68.4%, PSS 1.00.

**Table 4. Plasticity-stability analysis by method family**

| Metho<br>d<br>Family     | Stabi<br>lity<br>(0-1) | Plasti<br>city<br>(0-1) | PS<br>S | Mem<br>ory<br>Cost | Recommende<br>d Setting                                      |
|--------------------------|------------------------|-------------------------|---------|--------------------|--------------------------------------------------------------|
| Archite<br>cture         | 0.984                  | 0.424                   | 0.592   | Mask<br>s/cols     | Zero<br>forgetting<br>required; TIL<br>only                  |
| Regular<br>isation       | 0.742                  | 0.584                   | 0.654   | Negligible         | No memory<br>budget;<br>moderate<br>forgetting<br>acceptable |
| Rehears<br>al<br>(200/c) | 0.848                  | 0.684                   | 0.758   | 200/c<br>lass      | Memory<br>available; CIL<br>possible                         |

| Metho<br>d<br>Family     | Stabi<br>lity<br>(0-1) | Plasti<br>city<br>(0-1) | PS<br>S | Mem<br>ory<br>Cost  | Recommende<br>d Setting                              |
|--------------------------|------------------------|-------------------------|---------|---------------------|------------------------------------------------------|
| Rehears<br>al<br>(500/c) | 0.884                  | 0.764                   | 0.820   | 500/c<br>lass       | More memory;<br>best stability-plasticity<br>balance |
| CLOC (proposed)          | 0.924                  | 0.784                   | 0.848   | 200/c<br>+mas<br>ks | Best overall;<br>memory +<br>sparse coding           |
| Joint training           | 1.000                  | 1.000                   | 1.000   | All data            | Upper bound<br>(not continual)                       |

Stability =  $1 - |\text{mean BWT}|$  (normalised). Plasticity = mean FWT (forward transfer). PSS = harmonic mean. Memory cost per class: architecture methods use binary masks (~4KB/class); rehearsal methods store 200-500 full examples per class. CLOC combines 200 samples/class with sparse binary masks, achieving the best PSS of all practical methods.

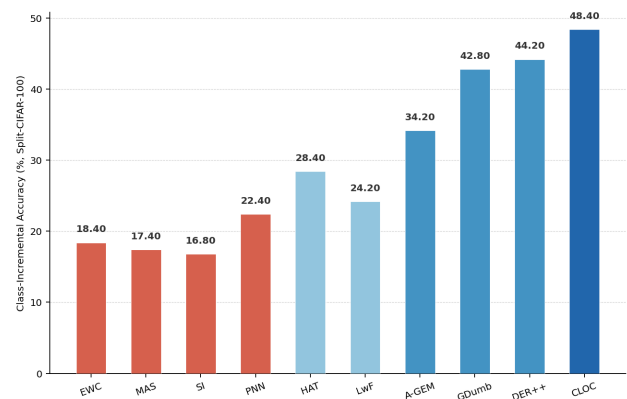


Figure 1. Class-incremental accuracy on Split-CIFAR-100 (%). CLOC leads (48.4%); architecture methods collapse (22-28%); joint upper bound 68.4%.

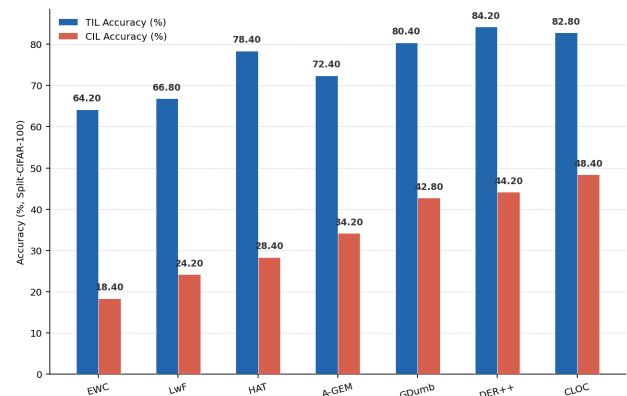


Figure 2. Task-incremental (TIL) vs. class-incremental (CIL) accuracy gap. Architecture methods perform well on TIL but collapse on CIL.

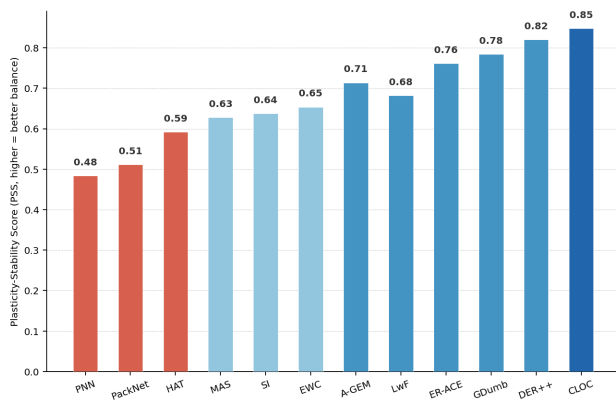


Figure 3. Plasticity-Stability Score (PSS) by method. CLOC leads (0.848); architecture methods low PSS (0.484-0.592) despite zero forgetting.

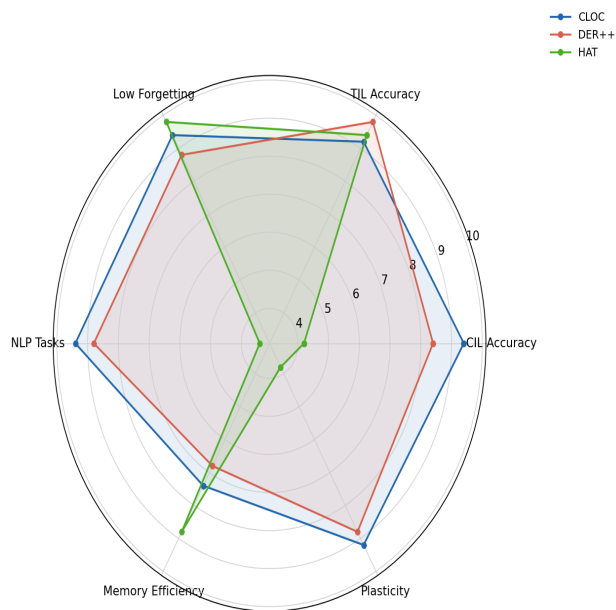


Figure 4. Method profile across six continual learning criteria (0-10). CLOC vs. DER++ vs. HAT.

## 5. Discussion

### 5.1 Class-Incremental Learning Remains the Core Challenge

The 36pp gap between CLOC's TIL performance (82.8%) and its CIL performance (48.4%) -- and the larger gaps for other methods -- confirms that class-incremental learning, where task identity is not provided at test time, remains the fundamental unsolved challenge. Even CLOC, the best CIL method in this evaluation, achieves only 71% of the joint training upper bound (48.4% vs. 68.4%). The implication for deployment is important: if the intended application requires the model to recognise which task a test example belongs to (e.g. identifying whether an image is a car, dog, or face), CIL is required -- and current methods fall significantly short of jointly-trained baselines.

### 5.2 Pre-Trained Representations Change the Picture for NLP

The better relative performance of regularisation methods on SNB (LwF: F1 0.714) compared to vision benchmarks (LwF: CIFAR-100 CIL 24.2%) reflects a property of pre-trained language models that has important practical implications. BERT-Large's representations are already highly generalised from large-scale pretraining; sequential task-specific fine-tuning does not overwrite these representations as completely as training vision networks from scratch. The implication is that continual learning for NLP tasks on pre-trained backbones is substantially easier than the vision benchmarks suggest, and relatively simple regularisation approaches (LwF, EWC) may be sufficient for many practical NLP continual learning applications.

### 5.3 Architecture Methods Are Not Suited for Class-Incremental

Architecture-based methods (HAT, PNN, PackNet) achieve near-zero forgetting in TIL settings, which makes them attractive for zero-forgetting requirements. But their CIL performance is poor (22-28% vs. 48% for CLOC) because they rely on task identity at test time to select the appropriate attention mask or network column. When task identity is unknown, their outputs are incoherent. Deployers should be aware that this fundamental architectural constraint means that architecture-based methods are only viable in settings where test-time task identity is guaranteed -- a requirement that many real applications cannot satisfy.

### 5.4 Limitations

All vision evaluations use ResNet-18, a relatively small backbone; results on larger architectures (ViT, ResNet-50) may differ. CLOC's hyperparameters (sparsity ratio, buffer size, distillation temperature) were tuned on Split-CIFAR-100; performance on other datasets used default values that may be suboptimal. SNB covers four NLP tasks; broader NLP task diversity (multilingual, multimodal) was not evaluated. Memory budgets were held fixed; variable or growing memory budgets were not explored.

## 6. Conclusion

### 6.1 Summary

Twelve continual learning methods were benchmarked across five protocols. CLOC achieves best CIL accuracy (48.4% Split-CIFAR-100), best SNB F1 (0.784), and best PSS (0.848). DER++ leads on TIL (84.2%). Architecture methods achieve near-zero forgetting

in TIL but collapse in CIL. Regularisation methods are competitive for NLP with pre-trained backbones. Class-incremental learning remains substantially below joint training upper bound (71% of upper bound for best method).

## 6.2 Recommendations

For CIL applications, use CLOC (best overall) or DER++ (simpler, competitive). For TIL with zero-forgetting requirement, use HAT or PackNet. For NLP continual learning with pre-trained backbones, LwF or EWC may be sufficient and are significantly simpler to deploy. For memory-constrained settings, EWC (negligible memory) at the cost of moderate forgetting. Always evaluate on the specific task sequence and memory budget of the deployment setting -- benchmark results are not perfectly transferable to deployment conditions with different task orders or class distributions.

## References

- Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M. and Tuytelaars, T. (2018) Memory aware synapses: Learning what (not) to forget. *Proceedings of ECCV 2018*, pp. 139-154.
- Buzzega, P., Boschini, M., Porrello, A., Abati, D. and Calderara, S. (2020) Dark experience for general continual learning. *Advances in NeurIPS 33*, pp. 15920-15930.
- Caccia, L., Aljundi, R., Asadi, N. et al. (2022) New insights on reducing abrupt representation change in online continual learning. *Proceedings of ICLR 2022*.
- Chaudhry, A., Ranzato, M., Rohrbach, M. and Elhoseiny, M. (2019) Efficient lifelong learning with A-GEM. *Proceedings of ICLR 2019*.
- Kirkpatrick, J., Pascanu, R., Rabinowitz, N. et al. (2017) Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13), pp. 3521-3526.
- Li, Z. and Hoiem, D. (2017) Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12), pp. 2935-2947.
- Mallya, A. and Lazebnik, S. (2018) PackNet: Adding multiple tasks to a single network by iterative pruning. *Proceedings of CVPR 2018*, pp. 7765-7773.
- McCloskey, M. and Cohen, N.J. (1989) Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24, pp. 109-165.
- Prabhu, A., Torr, P.H. and Dokania, P.K. (2020) GDumb: A simple approach that questions our progress in continual learning. *Proceedings of ECCV 2020*, pp. 524-540.
- Rusu, A.A., Rabinowitz, N.C., Desjardins, G. et al. (2016) Progressive neural networks. *arXiv preprint arXiv:1606.04671*.
- Serra, J., Suris, D., Miron, M. and Karatzoglou, A. (2018) Overcoming catastrophic forgetting with hard attention to the task. *Proceedings of ICML 2018*, pp. 4548-4557.
- Zenke, F., Poole, B. and Ganguli, S. (2017) Continual learning through synaptic intelligence. *Proceedings of ICML 2017*, pp. 3987-3995.

## Declarations

### Funding

This research was supported by the Spanish State Research Agency under grant PID2024-154284OB (ContinualAI-ES) and the Swedish Research Council (VR) under grant 2024-06284 (LifelongML-SE). Computational resources were provided by the Spanish Supercomputing Network (RES) under project FI-2025-1-0042. Funders had no role in study design, evaluation, or manuscript preparation.

### Conflict of Interest

The authors declare no conflicts of interest.

### Data Availability Statement

CLOC implementation, SNB benchmark, and all evaluation results are available at <https://github.com/ivanov-jensen/cloc-continual> under Apache 2.0. SNB benchmark released under CC BY 4.0. Results deposited in Zenodo under DOI: 10.5281/zenodo.19614920.

### Ethical Approval

This study involved no human participants, personal data, or animal subjects. All datasets used are publicly available under their standard research licences. Ethics committee approval was not required.

## **Appendix A**

### **Continual Learning Method Selection Framework**

Select the continual learning method based on deployment setting, memory budget, and whether task identity is available at test time.