ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

# Continual Unlearning through Memory Suppression

Alexander Krawczyk[1] and Alexander Gepperth[2]

University of Applied Sciences Fulda - Applied Computer Science
Leipziger Str. 123 - Fulda - Germany

**Abstract**. This study uncovers surprisingly effective synergies between the field of continual learning (CL) and machine unlearning (MUL). We extend the common class-incremental setting from CL to incorporate suppression requests in what we term class-incremental unlearning (CIUL). We present a light-weight approach to CIUL using replay/rehearsal-based CL approaches together with a selective replay strategy termed "Replay-To-Suppress" (RTS), where we actually *make use* of the catastrophic forgetting effect to achieve unlearning. In particular, we adapt a CL strategy termed adiabatic replay (AR) to achieve suppression at near-constant time complexity. We demonstrate excellent overall performance for all CL strategies extended by RTS on MNIST, F-MNIST and a latent encoded version of the challenging CIFAR and SVHN benchmarks.

## 1 Introduction

Machine Unlearning (MUL) is concerned with solutions for selective data removal from trained models. Nowadays, large-scale architectures can have up to billions of trainable parameters, which drastically increases the cost of retraining. In addition, the original training data may become inaccessible at the time of deletion requests [1]. This highlights the need for energy-efficient solutions to remove certain data from trained ML models. In this work, we propose an active forgetting mechanism termed *Replay-To-Suppress* (RTS) derived from conventional replay-based continual learning (CL) methods. We furthermore propose the scenario of class-incremental unlearning (CIUL), and adapt an approach termed *adiabatic replay* [2] to work in this scenario at constant time complexity. Our approach is based on 1) hippocampal-like replay of latent features from a pre-trained feature encoder, 2) using selective replay strategies, and 3) performing local updates so that only a near-constant time complexity is reached for model (re-)training.

## 2 Related Work

Exact unlearning is achievable by data sharding and the use of multiple temporary networks [3] or naive retraining. The vast majority of MUL techniques are only able to perform "approximate unlearning" [4, 5], for example by storing additional meta-data, such as gradients or parameters for future deletion requests. Weight scrubbing [6,7] adds noise tailored to the loss landscape using the Fisher Information Matrix (FIM). SCRUB(-R) [8] is based on a teacher-student

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

framework where a frozen model trains a student model with an alternating loss optimization similar to the min-max game of GANs. Continual Learning and Private Unlearning (CLPU) [9] is a CL-compliant framework where tasks are either marked as permanent, temporary or to be forgotten. Several sub-models are combined with experience replay (ER) to achieve exact unlearning on demand. The authors of [10] show an re-optimization-based approach to class-level selective forgetting in a task-incremental CL setup. This approach uses a data augmentation technique called "mnemonic codes" (a synthetic image generated once per class) embedded in each data sample combined with a special loss function. Selective Amnesia [11] frames the problem of active forgetting from the perspective of continual learning, combining conditional generative replay (GR) [12] with EWC [13].

## 3 Class-Incremental Unlearning

In the supervised Class-Incremental Unlearning (CIUL) scenario, a model parameterized by $\theta$ needs to continually learn from a sequence of training datasets $\mathcal{D}_1, \ldots, \mathcal{D}_k, \mathcal{D}_k = (x_k^i, y_k^i)_{i=1}^{n_k}$. For each task, we specify a dataset $\mathcal{D}_k^s \subset \mathcal{D}_k$ whose classes should be suppressed (**S**), and a complementary dataset $\mathcal{D}_k^r = \mathcal{D}_k \setminus \mathcal{D}_k^s$ whose classes should be learned (**L**). The classifier is expected to exhibit degraded performance or complete forgetting over suppressed classes when evaluated on a test set for $\mathcal{D}_k^s$ after the full training sequence.

## 4 Replay-To-Suppress

Replay of past data is a powerful paradigm to mitigate CF [14, 15]. RTS can be applied to virtually any replay-based technique by tweaking the replay mechanism to **exclude** samples of to-be-suppressed classes. This way, we use CF to remove unwanted information automatically, while at the same time protecting to-be-retained knowledge. For every new task, the RTS strategy is as follows:

1. generate samples either based on the incoming batch or conditionally based on classes "to-be-retained"
2. remove the generated samples belonging to classes "to-be-suppressed", with class identities being inferred by a forward pass of the model (this is an **optional step**, in case we don't use class-conditional sampling)
3. train the model on a batch composed of generated and incoming samples

## 5 Experiments

### 5.1 Implementation

All evaluated methods, except for AR, use the ANN structures given in Tab. 1. Training and testing is performed with a mini-batch size of 128, and if not stated otherwise, for 100 (MNIST/F-MNIST) and 200 (CIFAR/SVHN) epochs

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

respectively. The default optimizer is set to be ADAM with $\epsilon = 10^{-4}, \beta_1 = .9, \beta_2 = .999$.

| **Model architecture** | **Layer structure** |
|---|---|
| **DNN-1** (MNIST) | $D(400) \times 3 \rightarrow SM(10)$ |
| **DNN-2** (CIFAR/SVHN) | $D(800) \times 4 \rightarrow SM(10)$ |
| **CNN-1** (F-MNIST) | $C2D(32, 3 \times 3, 2 \times 2) \rightarrow MP(2 \times 2) \rightarrow$ $C2D(64, 3 \times 3, 2 \times 2) \rightarrow MP(2 \times 2) \rightarrow$ $D(512) \rightarrow D(256) \rightarrow SM(10)$ |
| **ENC-1** (MNIST/F-MNIST) | $D(512) \rightarrow D(256) \rightarrow D(128) \rightarrow D(2z)$ |
| **ENC-2** (CIFAR/SVHN) | $D(1024) \rightarrow D(512) \rightarrow D(256) \rightarrow D(2z)$ |

Table 1: ANN structures used in the empirical study. D=Dense, C2D=2D Convolution, MP=Max Pooling, SM=Softmax, RL=ReLU.

**Baseline**: An ANN jointly trained for 200 epochs on $\mathcal{D}^r = \bigcup_k \mathcal{D}_k^r$.

**Sequential Finetuning (SFT)**: The model is incrementally trained on each task $t_k$ from Seq. A/B Tab. 2, decreasing $\epsilon$ to $10^{-5}$ after $t_1$. Additional dropout with a rate of 0.3 is added to each dense layer.

**Elastic Weight Consolidation (EWC)** [13]: EWC is trained for 100 epochs with $\lambda = 100$. All available samples from $\mathcal{D}_k^r$ are used for FIM calculation, however, samples belonging to classes from $\mathcal{D}_k^s$ are excluded.

**Experience Replay (ER)** [14]: ER is trained via SGD with $\epsilon = 10^{-3}$ and uses reservoir sampling for episodic memory population. We remove samples from classes in $\mathcal{D}_k^s$ after each suppression task. A storage budget of 50 (MNIST/F-MNIST) and 100 (latent) samples per class is allocated. Additionally, we apply a sample-wise class-balanced loss weighting as proposed in [16].

**Deep Generative Replay (DGR)** [12]: DGR uses symmetric C-VAEs with $\beta = 1$, a latent dimension of $z = 25$ (MNIST), $z = 50$ (F-MNIST), and $z = 100$ (latent) as the encoder/decoder networks, see "ENC/DEC" from Tab. 1. The loss function is sigmoid cross-entropy, and the latent prior $p(z)$ follows a unit Gaussian distribution. The VAE is class-conditioned on the label space $p(y)$ to control image generation. To ensure balance, samples from the prior are generated in equal proportions, as demonstrated in [2, 16]. The DGR solver (DNN-1´) is trained using SGD with $\epsilon = 10^{-3}$ for 100 epochs. The generator is trained for 100 epochs on MNIST and F-MNIST, and 200 epochs on latent data.

**Selective Amnesia (SA)** [11]: This method uses the same C-VAE architecture as DGR but introduces a dedicated forgetting mechanism with a regularized loss that combines the corrupted, contrastive, and EWC losses. After each task, a frozen copy of the VAE $\hat{\theta}_{t_k}$ generates samples for FIM calculation. Similar to EWC, classes designated for suppression are excluded from generation by $\hat{\theta}_{t_k}$. SA forgetting is performed for $10^4$ steps using 5000 generated samples from $\hat{\theta}_{t_k}$. The EWC regularization strength, $\lambda_{EWC}$, is set to 100, while $\gamma_{SA}$ is kept at 1.0.

**Adiabatic Replay (AR)**: AR is used with $K = 100$ (MNIST), and $K = 256$ (F-MNIST/latent). The annealing control parameter $\sigma_{t_k}(t = 0)$ is set to $0.5\sigma_0$

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

(MNIST/F-MNIST) or $0.25\sigma_0$ (latent) for each task $t_k, k > 1$. Samples from classes to be suppressed are excluded from the set of samples generated by AR's variant generation by performing an additional forward pass after sampling.

### 5.2 Evaluation protocol

**Benchmark datasets** are MNIST, FashionMNIST, SVHN and CIFAR-10. Feature encoding is applied to SVHN and CIFAR data as described in [16]. Two CIUL sequences are constructed by splitting the dataset as shown in Tab. 2. Each CIUL experiment is run 10 times using a random initialization.

| **Sequence** | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ | $t_6$ | $t_7$ | $t_8$ |
|---|---|---|---|---|---|---|---|---|
| **CIUL-A** | L[0-5] | S[0-2] | L[6,7] | S[3,4] | L[8] | S[5] | L[9] | / |
| **CIUL-B** | L[0-3] | L[4-6] | S[4,5] | L[7] | S[6] | L[8] | S[7] | L[9] |

Table 2: CIUL sequences **A** and **B**. **L** stands for learn, **S** for suppress.

### 5.3 Measuring Continual Unlearning Performance

When denoting the test accuracy on task $i < j$ after training on task $j$ as $\alpha_{i,j}$, where $i < j$, suppression is defined as: $f^s = N^{-1} \sum_{i \in T^s} (\alpha_{i,i} - \alpha_{T,i})$. Retention $\alpha^r$ is measured as the classification accuracy on the union of all retention sets: $\alpha^r$ = accuracy on $\bigcup_{i=1}^{N} \mathcal{D}^{r_i}$.

## 6 Results

| | | **BASE** | **AR** | | **DGR** | | **ER** | | **SA** | | **EWC** | | **SFT** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Metric →** | $\alpha^r$ | $\alpha^r$ | $f^s$ | $\alpha^r$ | $f^s$ | $\alpha^r$ | $f^s$ | $\alpha^r$ | $f^s$ | $\alpha^r$ | $f^s$ | $\alpha^r$ | $f^s$ |
| **CIUL Sequence** | **MNIST-A** | .98 | .89 | .88 | .91 | .91 | **.92** | .85 | .84 | **.92** | .34 | .17 | .25 | .99 |
| | **MNIST-B** | .97 | .72 | .87 | .87 | .90 | **.92** | **.97** | .64 | .91 | .35 | .50 | .14 | .99 |
| | **FMNIST-A** | .96 | .79 | .58 | .92 | .87 | **.93** | .87 | .89 | **.90** | .60 | .54 | .24 | .98 |
| | **FMNIST-B** | .93 | .71 | .70 | .85 | .80 | **.91** | .80 | .72 | **.82** | .24 | .94 | .15 | .99 |
| | **SVHN-A** | .96 | .56 | .43 | **.95** | **.94** | **.95** | **.94** | .94 | **.94** | .20 | .93 | .20 | .98 |
| | **SVHN-B** | .94 | .62 | .54 | .91 | .90 | **.94** | .92 | .87 | **.93** | .08 | .97 | .08 | .99 |
| | **CIFAR-A** | .96 | .67 | .40 | .76 | .84 | **.94** | **.91** | .74 | .86 | .24 | .73 | .24 | .94 |
| | **CIFAR-B** | .82 | .40 | .41 | .79 | .52 | **.82** | **.80** | .81 | .55 | .15 | .92 | .15 | .96 |

Table 3: Performance comparison across CIUL task sequences, averaged over 10 runs. **Higher values indicate better performance** for both retention $\alpha^r$ and suppression $f^s$.

As can be seen from Tab. 3, all of the replay methods (ER, DGR, AR) enhanced by RTS (see Sec. 5.1) show strong performance overall. DGR has to use

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

an increasing number of generated samples for each task, which can be problematic (see [16]), something AR does not have to do, although its performance is slightly inferior. ER is a very strong baseline for CL, but one might expect that the limited replay budget will lead to problems for CIUL sequences with a larger number of tasks. SA performs strong suppression, but lacks the ability to retain knowledge. It also requires additional training time since forgetting does not occur simultaneously with learning new data. EWC gives variable results: sometimes it has problems with suppression and/or retention, this depends heavily on the EWC balance parameter. SFT only adapts to the most recent task, as expected, and fails completely at retention.

## 7 Discussion

**Synergies between CL and MUL** Most MUL methods are disconnected from the field of CL. In general, they either require access to the complete forget set, retain set or both, see e.g., [4, 6, 7, 17–20]. The few exceptions either use multi-headed classifiers [10] or are incompatible with CL since the task sequence needs to be known beforehand [9]. It is quite surprising that the RTS technique we propose here, a straightforward modification to off-the-shelf CL approaches, can achieve such high-quality results on simple and complex CIUL benchmarks, see Tab. 3. Rather than trying to avoid it at all cost, we stress that RTS uses catastrophic forgetting (CF) as a tool for enabling *controlled* suppression.

**Suppression performance/efficiency in RTS** We speculate that suppression is more effective for less recent classes, since they are more prone to suffer from transient CF. Data complexity clearly influences suppression, since more complex data should suffer from CF more strongly. However, RTS is remarkably efficient, especially for inherently sample-efficient techniques like AR, enabling its use with really large-scale models. In contrast, dedicated forgetting operations described in MUL are often inefficient and increase storage and computational costs [11].

**A note on methods relying on regularization** It is common to apply a penalty term to the loss like EWC and LwF in MUL [10, 11]. However, these approaches require a-priori knowledge of the task (suppression) sequence because they contain parameters that can only be tuned "in hindsight" by cross-validation. Furthermore, the order of learning and suppression has to be known beforehand to allow proper exclusion of classes from FIM calculation.

## 8 Conclusion and future work

CL methods may benefit from the inclusion of active forgetting techniques due to capacity or memory constraints, as knowledge already acquired may become outdated over time, requiring the removal of data that does not actively contribute to solving the objective of the deployed model. In addition, consolidated data may contain harmful information that needs to be removed to avoid malfunctions. We have shown that RTS enables several CL techniques to be effective in selectively unlearning information. A key point of future research should be

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

the *scaling* of CIUL to a very large number of tasks and samples, where a closer study of constant-time replay approaches such as AR could be beneficial, as the linear scaling of DGR and the unbounded growth of the required replay budget for ER on a large number of tasks/classes limit their practical applicability.

## References

[1] Jinu Gong, Joonhyuk Kang, Osvaldo Simeone, and Rahif Kassab. Forget-svgd: Particle-based bayesian federated unlearning. In *2022 IEEE Data Science and Learning Workshop (DSLW)*, pages 1–6. IEEE, 2022.

[2] Alexander Krawczyk and Alexander Gepperth. Adiabatic replay for continual learning. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10, 2024.

[3] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

[4] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11516–11524, 2021.

[5] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*. PMLR, 2021.

[6] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9304–9312, 2020.

[7] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations, 2020.

[8] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning, 2023.

[9] Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pages 243–254. PMLR, 2022.

[10] Takashi Shibata, Go Irie, Daiki Ikami, and Yu Mitsuzumi. Learning with selective forgetting. In *IJCAI*, volume 3, page 4, 2021.

[11] Alvin Heng and Harold Soh. Selective amnesia: A continual learning approach to forgetting in deep generative models, 2023.

[12] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *arXiv preprint arXiv:1705.08690*, 2017.

[13] James Kirkpatrick and et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13):3521–3526, 2017.

[14] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32:350–360, 2019.

[15] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *CVPR*, pages 8250–8259, 2021.

[16] Alexander Krawczyk and Alexander Gepperth. An analysis of best-practice strategies for replay and rehearsal in continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4196–4204, 2024.

[17] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.

[18] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[19] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Variational bayesian unlearning. *Advances in Neural Information Processing Systems*, 33:16025–16036, 2020.

[20] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, pages 2008–2016. PMLR, 2021.