BrAIn Lab.
Brain and Artificial Intelligence Lab.

SEOULTECH
SEOUL NATIONAL UNIVERSITY OF SCIENCE & TECHNOLOGY

Online Continual Learning with Maximally Interfered Retrieval

Rahaf Aljundi et al.

2026.01.26

BrainLAB Journal Club

Overview

01 Author

BrAIn Lab.
Brain and Artificial Intelligence Lab.

Rahaf Aljundi

FOLLOW

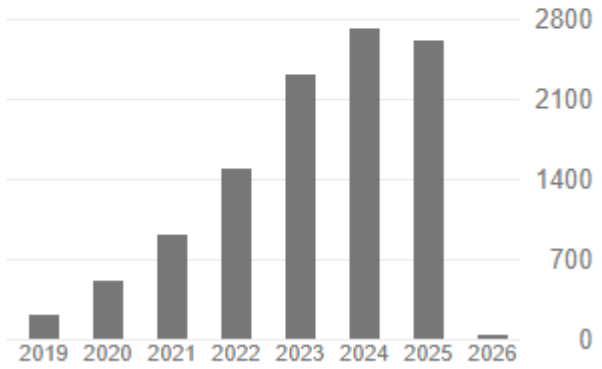
Senior Researcher at Toyota Motor Europe
Verified email at toyota-europe.com - Homepage

Machine learning Computer vision

TITLE	CITED BY	YEAR
A continual learning survey: Defying forgetting in classification tasks M De Lange, R Aljundi, M Masana, S Parisot, X Jia, A Leonardis, ... IEEE transactions on pattern analysis and machine intelligence 44 (7), 3366-3385	3014	2021
Memory aware synapses: Learning what (not) to forget R Aljundi, F Babiloni, M Elhoseiny, M Rohrbach, T Tuytelaars Proceedings of the European conference on computer vision (ECCV), 139-154	2388	2018
Gradient based sample selection for online continual learning R Aljundi, M Lin, B Goujaud, Y Bengio Advances in neural information processing systems 32	1227	2019
Expert gate: Lifelong learning with a network of experts R Aljundi, P Chakravarty, T Tuytelaars Proceedings of the IEEE conference on computer vision and pattern ...	966	2017
Online continual learning with maximal interfered retrieval R Aljundi, E Belilovsky, T Tuytelaars, L Charlin, M Caccia, M Lin, ... 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.	799	2019

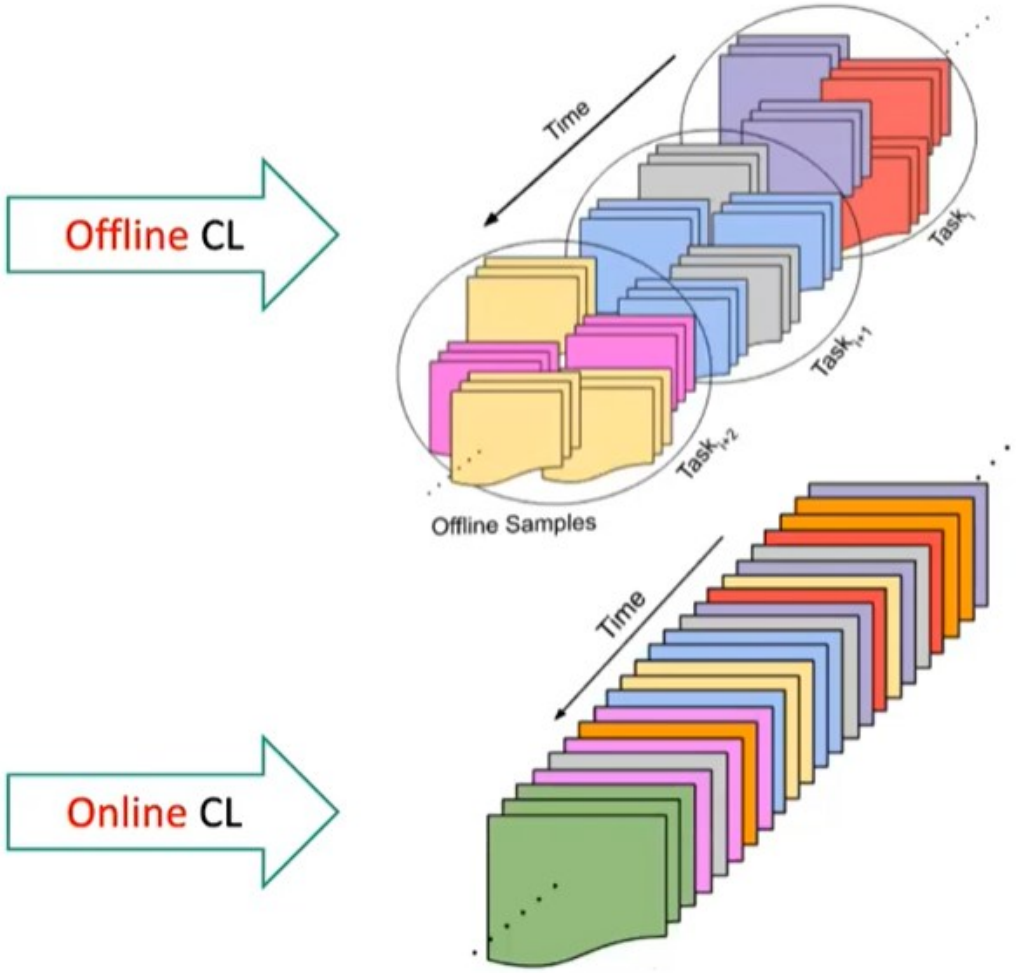
Cited by VIEW ALL

	All	Since 2021
Citations	11095	10139
h-index	21	20
i10-index	29	27



BrAIn Lab.
Brain and Artificial Intelligence Lab.

Background study



- Artificial neural networks have **exceeded human-level performance in accomplishing individual narrow tasks**. However, such success remains limited compared to human intelligence that can continually **learn and perform an unlimited number of tasks**.
- **Continual learning**, the setting where a learning agent is faced with **a never ending stream of data**, continues to be a great challenge for modern machine learning systems.
- In particular the **online or "single-pass through the data" setting** has gained attention as a natural setting that is difficult to tackle.

Mitigate forgetting

- **Replay method**: previous knowledge is stored either directly in a replay buffer, or compressed in a generative model.
- **Highlight**: The loss of some previous samples may be unaffected or even improved, thus retraining on them is wasteful.
- **Research question**: **what samples should be replayed from the previous history** when new samples are received.
- **Proposed**: Instead of using randomly selected or generated samples from the previous history, we find samples that would be **(maximally) interfered** by the new incoming sample(s).

1) Problem Space

- We consider a (potentially infinite) stream of data where **at each time step, t**, the system **receives a new set of samples X_t, Y_t** drawn non i.i.d from **a current distribution D_t**.
- We aim to learn **a classifier f** parameterized by **θ** that **minimizes a predefined loss L** on new sample(s) from the data stream **without increasing the loss on previously observed samples**.

2) Maximally Interfered Sampling from a Replay Memory (Experience Replay)

- In this approach the learner is allocated **a memory M** of finite size, which is updated by the use of **sampling** as the stream of samples arrives.
- Standard objective: $\min_{\theta} \mathcal{L}(f_\theta(\boldsymbol{X_t}), \boldsymbol{Y}_t)$
- Parameters update from the incoming batch: $\theta^v = \theta - \alpha \nabla \mathcal{L}(f_\theta(\boldsymbol{X}_t), \boldsymbol{Y}_t)$
- Search for the top-k values x ∈ M : $s_{MI\text{-}1}(x) = l(f_{\theta^v}(x), y) - l(f_\theta(x), y)$
- We may also augment the memory to store best loss observed so far for that sample: $l(f_{\theta^*}(x), y)$

$$s_{MI\text{-}2}(x) = l(f_{\theta^v}(x), y) - \min\left(l(f_\theta(x), y), l(f_{\theta^*}(x), y)\right)$$

2) Maximally Interfered Sampling from a Replay Memory

- Denote the budget of samples to retrieve as B.
- **To encourage diversity** they perform an initial random sampling of the memory, selecting C samples where C > B before applying the search criterion.
- This also **reduces the compute cost of the search.**

Algorithm 1: Experience MIR (ER-MIR)

Input: Learning rate α, Subset size C; Budget $\mathcal{B}$

Initialize: Memory $\mathcal{M}$; θ
for $t \in 1..T$ **do**
  for $B_n \sim D_t$ **do**
    %%Virtual Update
    $\theta^v \leftarrow \mathrm{SGD}(B_n, \alpha)$
    %Select C samples
    $B_C \sim \mathcal{M}$
    %Select based on score
    $S \leftarrow sort(s_{MI}(B_C))$
    $B_{\mathcal{M}_C} \leftarrow \{S_i\}_{i=1}^{\mathcal{B}}$
    $\theta \leftarrow SGD(B_n \cup B_{\mathcal{M}_C}, \alpha)$
    %Add samples to memory
    $\mathcal{M} \leftarrow UpdateMemory(B_n)$;
  end
end

3) Maximally Interfered Sampling from a Generative Model

- We want to find in the given feature space **data points that maximize the difference between their loss before and after the estimated parameters update**:

$$\max_{\boldsymbol{Z}} \mathcal{L}\big(f_{\theta^v}(g_\gamma(\boldsymbol{Z})), \boldsymbol{Y}^*\big) - \mathcal{L}\big(f_{\theta'}(g_\gamma(\boldsymbol{Z})), \boldsymbol{Y}^*\big) \quad \text{s.t.} \quad ||z_i - z_j||_2^2 > \epsilon \, \forall z_i, z_j \in \boldsymbol{Z} \text{ with } z_i \neq z_j$$

- $\boldsymbol{\epsilon}$ a threshold to encourage the diversity of the retrieved points.
- From these points we reconstruct the full corresponding input samples $\boldsymbol{X}^{'} = g_\gamma(Z)$ and use them to estimate the new parameters update $\min_\theta \mathcal{L}(f_\theta(\boldsymbol{X}_t \cup \boldsymbol{X}^{'}))$

3) Maximally Interfered Sampling from a Generative Model

- We also need an estimate of the label y∗ when using a generator.
- The predicted labels are given by $f_{\theta'}$ as pseudo labels to estimate y∗

$$y_{pre} = f_{\theta'}(g_\gamma(z)) \qquad \hat{y} = f_{\theta^v}(g_\gamma(z))$$

- Generative models are known to generate blurry images and images with mix of categories.
- To avoid such a source of noise in the optimization, they **minimize an entropy penalty to encourage generating points for which the previous model is confident**.
- The final objective of the generator-based retrieval is

$$\max_{\boldsymbol{Z}} \sum_{z \in \boldsymbol{Z}} [D_{KL}(y_{pre} \parallel \hat{y}) - \alpha H(y_{pre})] \quad s.t. \quad ||z_i - z_j||_2^2 > \epsilon \; \forall z_i, z_j \in \boldsymbol{Z} \text{ with } z_i \neq z_j$$

03 Proposed Method

4) A Hybrid Approach

- **Training generative models in the continual learning setting** on more challenging datasets like CIFAR10 **remains an open research problem**.
- **Storing samples** for replay is also problematic as it is **constrained by storage costs** and very-large memories can become difficult to search.
- To leverage the benefits of both, they use a hybrid approach where **an autoencoder is first trained offline to store and compress incoming memories.**
- By storing lightweight representations, the buffer can store more data for the same fixed amount of memory.

Algorithm 3: AE-MIR

Input: Learning rate α, Subset size C; Budget $\mathcal{B}$, Gen. Epochs N_{gen}

Initialize: Memory $\mathcal{M}$; θ, θ_{ae}
for $t \in 1..T$ **do**
  %%Offline Generator Training
  for $epoch \in 1...N_{gen}$ **do**
    for $B_n \sim D_t$ **do**
      $h \leftarrow \texttt{Encode}(\theta_{ae}; B_n)$
      $\tilde{B}_n \leftarrow \texttt{Decode}(\theta_{ae}; h)$
      $\text{loss}_{ae} \leftarrow \text{MSE}(\tilde{B}_n, B_n)$
      Adam $(\text{loss}_{ae}, \theta_{ae})$
    end
  end
  for $B_n \sim D_t$ **do**
    %%Virtual Update
    $\theta_v \leftarrow \texttt{SGD}(B_n, \alpha)$
    %%Autoencode batch
    $h \leftarrow \texttt{Encode}(\theta_{ae}; B_n)$
    $\tilde{B}_n \leftarrow \texttt{Decode}(\theta_{ae}; h)$
    %Select C samples
    $B_{\mathcal{C}} \sim \mathcal{M}$
    $B_G \leftarrow$ Retrieve samples acc. to Eq 1
    %%Store compressed rep.
    $\mathcal{M} \leftarrow$ UpdateMemory(h,L_n)
    %% Train the Classifier
    $\theta \leftarrow SGD(\tilde{B}_n \cup B_{\mathcal{M}_C}, \alpha)$
  end
end

They evaluate the proposed method under the **generative and experience replay** settings.

- **MNIST Split**: splits MNIST data to create 5 different tasks with non-overlapping classes. They consider the setting with 1000 samples per task.
- **Permuted MNIST**: permutes MNIST to create 10 different tasks.
- **CIFAR-10 Split**: splits CIFAR-10 dataset into 5 disjoint tasks with 9,750 samples per task and 250 retained for validation.
- **MiniImagenet Split**: splits MiniImagenet dataset into 20 disjoint tasks with 5 classes each.

1) Experience Replay (ER)

	Accuracy ↑	Forgetting ↓
iid online	86.8 ± 1.1	N/A
iid offline	92.3 ± 0.5	N/A
fine-tuning	19.0 ± 0.2	97.8 ± 0.2
GEN	79.3 ± 0.6	19.5 ± 0.8
GEN-MIR	$\mathbf{82.1 \pm 0.3}$	$\mathbf{17.0 \pm 0.4}$
GEM [27]	86.3 ± 1.4	11.2 ± 1.2
ER	82.1 ± 1.5	15.0 ± 2.1
ER-MIR	$\mathbf{87.6 \pm 0.7}$	$\mathbf{7.0 \pm 0.9}$

	Accuracy ↑	Forgetting ↓
iid online	73.8 ± 1.2	N/A
iid offline	86.6 ± 0.5	N/A
fine-tuning	64.6 ± 1.7	15.2 ± 1.9
GEN	79.7 ± 0.1	5.8 ± 0.2
GEN-MIR	$\mathbf{80.4 \pm 0.2}$	$\mathbf{4.8 \pm 0.2}$
GEM [27]	78.8 ± 0.4	$\mathbf{3.1 \pm 0.5}$
ER	78.9 ± 0.6	3.8 ± 0.6
ER-MIR	$\mathbf{80.1 \pm 0.4}$	3.9 ± 0.3

Table 1: Results for MNIST SPLIT (left) and Permuted MNIST (right). We report the Average Accuracy (higher is better) and Average Forgetting (lower is better) after the final task. We split results into privileged baselines, methods that don't use a memory storage, and those that store memories. For the ER methods, 50 memories per class are allowed. Each approach is run 20 times.

1) Experience Replay (ER)

	Accuracy ↑			Forgetting ↓		
	$M = 20$	$M = 50$	$M = 100$	$M = 20$	$M = 50$	$M = 100$
iid online	60.8 ± 1.0	60.8 ± 1.0	60.8 ± 1.0	N/A	N/A	N/A
iid offline	79.2 ± 0.4	79.2 ± 0.4	79.2 ± 0.4	N/A	N/A	N/A
GEM [27]	16.8 ± 1.1	17.1 ± 1.0	17.5 ± 1.6	73.5 ± 1.7	70.7 ± 4.5	71.7 ± 1.3
iCarl (5 iter) [31]	28.6 ± 1.2	33.7 ± 1.6	32.4 ± 2.1	$\mathbf{49 \pm 2.4}$	40.6 ± 1.1	40 ± 1.8
fine-tuning	18.4 ± 0.3	18.4 ± 0.3	18.4 ± 0.3	85.4 ± 0.7	85.4 ± 0.7	85.4 ± 0.7
ER	27.5 ± 1.2	33.1 ± 1.7	41.3 ± 1.9	50.5 ± 2.4	35.4 ± 2.0	23.3 ± 2.9
ER-MIR	$\mathbf{29.8 \pm 1.1}$	$\mathbf{40.0 \pm 1.1}$	$\mathbf{47.6 \pm 1.1}$	50.2 ± 2.0	$\mathbf{30.2 \pm 2.3}$	$\mathbf{17.4 \pm 2.1}$

Table 2: CIFAR-10 results. Memories per class M, we report (a) Accuracy, (b) Forgetting (lower is better). For larger sizes of memory ER-MIR has better accuracy and improved forgetting metric. Each approach is run 15 times.

1) Experience Replay (ER)

	Number of iterations	
	1	5
iid online	60.8 ± 1.0	62.0 ± 0.9
ER	41.3 ± 1.9	42.4 ± 1.1
ER-MIR	47.6 ± 1.1	49.3 ± 0.1

Table 3: CIFAR-10 accuracy (↑) results for increased iterations and 100 memories per class. Each approach is run 15 times.

	Accuracy ↑	Forgetting ↓
ER	24.7 ± 0.7	23.5 ± 1.0
ER-MIR	25.2 ± 0.6	18.0 ± 0.8

Table 4: MinImagenet results. 100 memories per class and using 3 updates per incoming batch, accuracy is slightly better and forgetting is greatly improved. Each approach is run 15 times

2) Generative Replay (GEN)

	Accuracy ↑	Forgetting ↓
iid online	86.8 ± 1.1	N/A
iid offline	92.3 ± 0.5	N/A
fine-tuning	19.0 ± 0.2	97.8 ± 0.2
GEN	79.3 ± 0.6	19.5 ± 0.8
GEN-MIR	$\mathbf{82.1 \pm 0.3}$	$\mathbf{17.0 \pm 0.4}$
GEM [27]	86.3 ± 1.4	11.2 ± 1.2
ER	82.1 ± 1.5	15.0 ± 2.1
ER-MIR	$\mathbf{87.6 \pm 0.7}$	$\mathbf{7.0 \pm 0.9}$

	Accuracy ↑	Forgetting ↓
iid online	73.8 ± 1.2	N/A
iid offline	86.6 ± 0.5	N/A
fine-tuning	64.6 ± 1.7	15.2 ± 1.9
GEN	79.7 ± 0.1	5.8 ± 0.2
GEN-MIR	$\mathbf{80.4 \pm 0.2}$	$\mathbf{4.8 \pm 0.2}$
GEM [27]	78.8 ± 0.4	$\mathbf{3.1 \pm 0.5}$
ER	78.9 ± 0.6	3.8 ± 0.6
ER-MIR	$\mathbf{80.1 \pm 0.4}$	3.9 ± 0.3

Table 1: Results for MNIST SPLIT (left) and Permuted MNIST (right). We report the Average Accuracy (higher is better) and Average Forgetting (lower is better) after the final task. We split results into privileged baselines, methods that don't use a memory storage, and those that store memories. For the ER methods, 50 memories per class are allowed. Each approach is run 20 times.

2) Generative Replay (GEN)

	MNIST Split	Permuted MNIST
GEN	107.2 ± 0.2	196.7 ± 0.7
GEN-MIR	$\mathbf{102.5 \pm 0.2}$	$\mathbf{193.7 \pm 1.0}$

Table 5: Generator's loss (↓), i.e. negative ELBO, on the MNIST datasets. Our methodology outperforms the baseline in online continual generative modeling as well.

Previous study found that generative replay **is not yet a viable strategy for CIFAR-10** given the current state of the generative modeling. Came to the same conclusion, it led to design of the hybrid approach.

BrAIn Lab.
Brain and Artificial Intelligence Lab.

3) Hybrid Approach

The classifier tend to classify all real samples as belonging to the classes of the last task, yielding low test accuracy. To address this problem, they **first autoencode the incoming data with the generator before passing it to the classifier**.

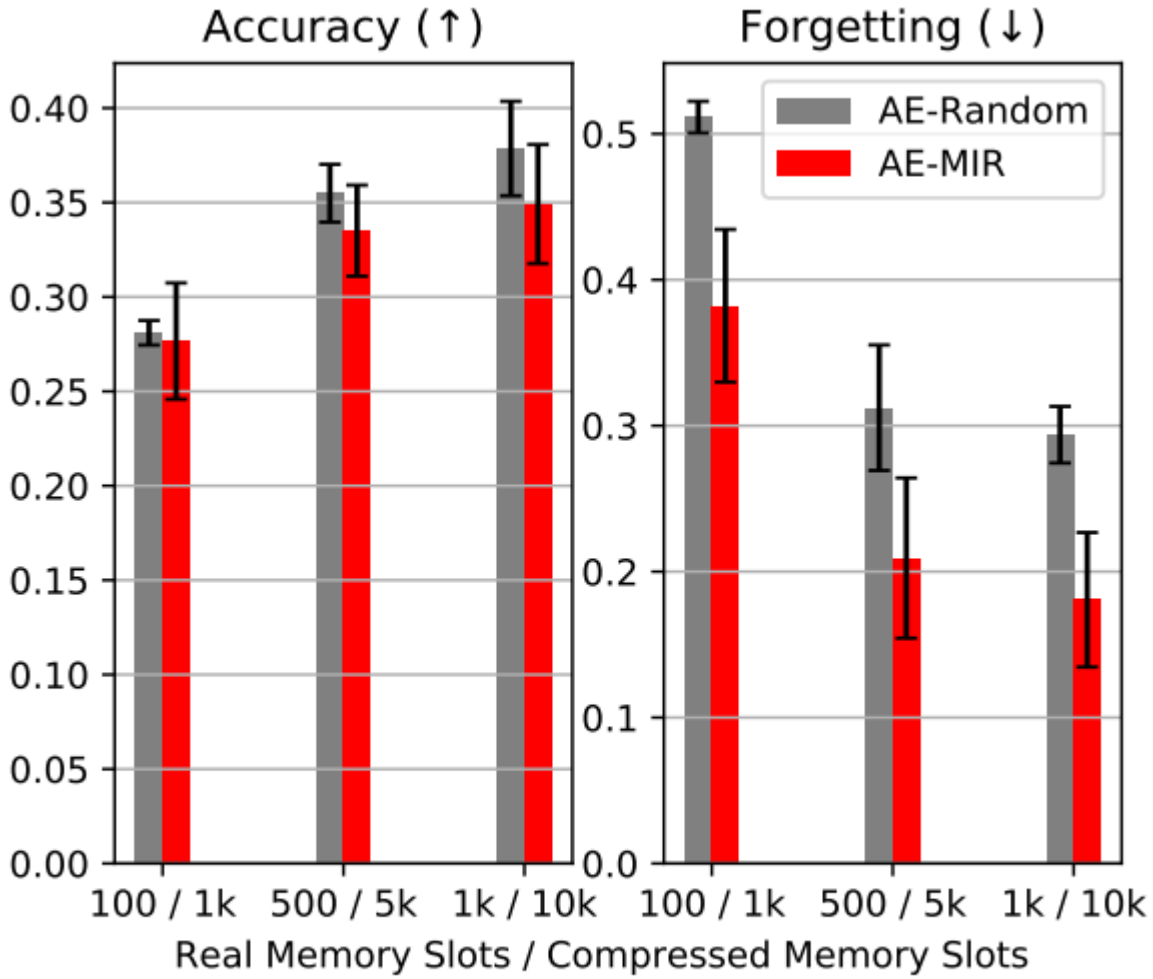


Figure 4: Results for the Hybrid Approach

05 Conclusion

BrAIn Lab.
Brain and Artificial Intelligence Lab.

- They have proposed **a criterion for retrieving relevant memories** in an online continual learning setting.
- Retrieving interfered samples **reduces forgetting** and significantly improves on random sampling and standard baselines.
- The results and analysis also shed light on **the feasibility and challenges of using generative modeling** in the online continual learning setting.

Thank You

2026.01.26

BrainLAB Journal Club