# How Relevant is Selective Memory Population in Lifelong Language Learning?

**Vladimir Araujo[1,2], Helena Balabin[1], Julio Hurtado[3], Alvaro Soto[2], Marie-Francine Moens[1]**
[1]KU Leuven, [2]Pontificia Universidad Católica de Chile, [3]University of Pisa
vgaraujo@uc.cl, helena.balabin@kuleuven.be,
julio.hurtado@di.unipi.it, asoto@ing.puc.cl, sien.moens@kuleuven.be

## Abstract

Lifelong language learning seeks to have models continuously learn multiple tasks in a sequential order without suffering from catastrophic forgetting. State-of-the-art approaches rely on sparse experience replay as the primary approach to prevent forgetting. Experience replay usually adopts sampling methods for the memory population; however, the effect of the chosen sampling strategy on model performance has not yet been studied. In this paper, we investigate how relevant the selective memory population is in the lifelong learning process of text classification and question-answering tasks. We found that methods that randomly store a uniform number of samples from the entire data stream lead to high performances, especially for low memory size, which is consistent with computer vision studies.

## 1 Introduction

While humans learn throughout their lifetime, current deep learning models are restricted to a bounded environment, where the input distribution is fixed. When those models are sequentially learning new tasks, they suffer from catastrophic forgetting (McCloskey and Cohen, 1989; Ratcliff, 1990) because the input distribution changes.

Several methods have been proposed to address catastrophic forgetting, mainly for computer vision (CV) (Delange et al., 2021) and few others for natural language processing (NLP) (Biesialska et al., 2020). In both, one of the prominent approaches is experience replay with episodic memory (Hayes et al., 2021), which aims to store previously seen training examples and later use them to perform gradient updates while training on new tasks.

In the experience replay approach, random sampling is the de facto method for the memory population, as it has shown good results in CV (Chaudhry et al., 2019; Wu et al., 2019; Hayes et al., 2020). In contrast, other works have shown that memory selection is relevant for deep reinforcement learning (Isele and Cosgun, 2018), image classification (Chaudhry et al., 2018; Sun et al., 2022), and analogical reasoning (Hayes and Kanan, 2021). However, no previous work has explored NLP tasks, which raises the question of whether memory selection is necessary for lifelong language learning.

In this paper, we adopt and evaluate seven memory population methods under a lifelong language learning setup with sparse experience replay. We conducted experiments with text classification and question answering tasks. We find that methods that obtain memory with a random sample from the global data distribution for text classification provide the best results in both high and low memory regimes. Conversely, for the question answering task, a method that provides a balanced memory composition per task performs better.

## 2 Related Work

**Lifelong Learning in NLP.** Rather than training a language model on a fixed dataset, lifelong (continual) language learning setups consist of a stream of tasks (e.g., text classification). In this setup, a model aims to retain the most relevant information to prevent catastrophic forgetting. Existing approaches for NLP include purely replay-based methods (d'Autume et al., 2019; Han et al., 2020; Araujo et al., 2022), meta-learning based methods (Wang et al., 2020; Holla et al., 2020) and generative replay-based methods (Sun et al., 2020a,b).

**Memory Selection in Lifelong Learning.** Several strategies have been proposed to store and select the most relevant training examples in memory. Early work has shown that reservoir sampling prevents catastrophic forgetting in lifelong reinforcement learning (Isele and Cosgun, 2018) and supervised learning (Chaudhry et al., 2019) with limited memory. More recent works have explored criteria-based selection methods, showing that maximum-



*Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 154–160
November 20–23, 2022. 

loss examples are helpful for analogical reasoning (Hayes and Kanan, 2021) and gradient-based (Aljundi et al., 2019) or information-theoretic (Sun et al., 2022) selection for image classification.

## 3 Lifelong Language Learning Setup

We consider the lifelong language learning setting proposed by d'Autume et al. (2019), in which a model learns multiple tasks in sequential order from a stream of training examples[1]. In this setup, each example is only allowed to be viewed once.

This setup adopts sparse experience replay, which performs a gradient update at a certain interval during training. We leverage this method, as d'Autume et al. (2019) have shown that a sparse 1% rate of replaying to learning new examples is sufficient for lifelong language learning.

This setting also includes local adaptation (Sprechmann et al., 2018), which is a process that retrieves K-nearest neighbors examples from memory to update model parameters used to predict a particular test example. However, recent works have tried to reduce its use (Wang et al., 2020) or even avoid it (Holla et al., 2020) because it significantly slows down the inference speed. We do not use this mechanism in our main experimentation because our goal is to analyze the effect of selective memory on the generalization of the model. Nevertheless, Section 6 briefly shows how resulting memory composition influences local adaptation.

## 4 Selective Episodic Memory

For the previously described lifelong learning setup, we extend a replay model (see Section 5) with the following seven memory population methods:

**Naive Random.** A basic method for memory population. It samples a percentage of elements of each task. In our experiments, the percentage value is the same as the memory capacity, and we sample the elements on the fly from the current batch.

**Reservoir.** A reservoir (Vitter, 1985) allows sampling elements from a stream without knowing how many elements to expect. It samples each element with a probability $\frac{M}{N}$ where $N$ is the number of elements observed so far and $M$ is the memory size. This way, it acts randomly to maintain a uniform sample from the already seen stream.

[1]We use an available implementation of this setup: https://github.com/vgaraujov/LLL-NLP

**Ring Buffer.** Similar to Lopez-Paz and Ranzato (2017), this method allocates $\frac{M}{C}$ elements for each class $C$ of the task in memory. The strategy is a FIFO buffer, so the memory is always filled with the latest task observations. If the total number of classes is unknown, the value of $M$ is gradually reduced as new tasks are observed.

**Surprise.** Unexpected events have been shown to influence episodic memory in humans (Cheng and Frank, 2008). One way to measure surprise is by computing the entropy of the output distribution of an input batch. Analogous to Isele and Cosgun (2018), we use the time difference between the current entropy value and that of the previous batch to sample high-surprise elements.

**Minimum Margin.** Similar to Hayes and Kanan (2021), who introduced a margin-based method for CV replay models, we define the margin as the difference between the probability of the true class and the probability of the other most likely class. We store the most uncertain examples, that is, those with the smallest margin for which the probability of the true class is only marginally different from the probability of the other most likely class.

**Maximum Loss.** Analogous to the previous strategy, the maximum loss strategy aims to store samples with high uncertainty. However, this time it is based on storing samples with a high loss value (Hayes and Kanan, 2021). Here, we slightly modify the strategy by evaluating the loss for an entire batch, therefore storing and overriding whole batches in memory.

**Mean of Features (MoF).** Similar to Rebuffi et al. (2017); Chaudhry et al. (2019), we calculate the average feature vector based on averaging the final `[CLS]` representations in memory for a given class. If the representation of an input example has a smaller distance to its average feature vector than the entry in the memory with the largest distance to the average, we store the new incoming example and update the respective average feature vector.

## 5 Experimental Setup

**Datasets.** We adopt the evaluation methodology and datasets proposed by (d'Autume et al., 2019).

For text classification, we use five datasets from (Zhang et al., 2015): AGNews classification, Yelp sentiment analysis, Amazon sentiment analysis, DBPedia article classification and Yahoo questions

| Order | N. Random | Reservoir | Ring Buffer | Surprise | Min. Margin | Max. Loss | MoF |
|---|---|---|---|---|---|---|---|
| Text Classification (Accuracy) | | | | | | | |
| i. | 70.88±1.22 | 69.54±5.99 | 68.36±3.61 | 53.74±1.83 | 71.40±0.83 | 56.59±1.61 | 60.34±7.39 |
| ii. | 72.17±0.41 | 73.41±1.14 | 74.32±0.35 | 69.40±2.14 | 71.68±1.32 | 70.82±2.62 | 65.62±4.87 |
| iii. | 65.37±1.32 | 67.79±1.34 | 65.13±2.29 | 63.00±2.44 | 63.35±0.69 | 67.64±0.96 | 56.98±2.46 |
| iv. | 72.72±0.79 | 73.32±0.89 | 69.99±2.35 | 57.46±2.97 | 72.29±1.02 | 59.63±2.25 | 63.30±1.31 |
| avg. | 70.29±0.94 | 70.99±2.34 | 69.45±2.15 | 60.90±2.35 | 69.68±0.96 | 63.67±1.86 | 61.56±4.01 |
| Question Answering (F1 score) | | | | | | | |
| i. | 59.32±1.12 | 59.34±0.73 | 59.12±0.63 | 61.24±0.08 | 59.24±1.03 | 59.40±1.06 | 59.42±0.42 |
| ii. | 58.40±1.22 | 58.99±0.53 | 59.38±0.26 | 59.51±0.44 | 58.48±0.67 | 59.62±0.64 | 57.06±0.95 |
| iii. | 52.95±1.44 | 53.47±0.51 | 54.61±0.78 | 50.10±0.64 | 53.02±0.64 | 44.77±1.04 | 50.37±3.81 |
| iv. | 60.56±0.76 | 60.03±0.18 | 60.49±0.62 | 61.00±0.39 | 59.93±0.69 | 60.16±0.48 | 59.69±0.47 |
| avg. | 57.81±1.13 | 57.96±0.48 | 58.40±0.57 | 57.96±0.39 | 57.67±0.76 | 55.99±0.80 | 56.63±1.41 |

Table 1: Summary of results for text classification and question answering using sparse experience replay and selective episodic memory population approaches. We report the mean accuracy or F1 score as well as the respective standard deviation across five runs with different random seeds.

and answers categorization. Both sentiment analysis tasks share the same labels. In total, we obtain 575,000 training and 38,000 test examples with 33 classes from all datasets using four task orders:

(i) Yelp → AGNews → DBPedia → Amazon → Yahoo
(ii) DBPedia → Yahoo → AGNews → Amazon → Yelp
(iii) Yelp → Yahoo → Amazon → DBpedia → AGNews
(iv) AGNews → Yelp → Amazon → Yahoo → DBpedia

For question answering, we use the following three datasets: SQuAD 1.1 (Rajpurkar et al., 2016), QuAC (Choi et al., 2018), and TriviaQA (Joshi et al., 2017). The latter has two sections, Web and Wikipedia, which we consider separate datasets. We obtain 60,000-90,000 training and 7,000-10,000 validation examples per task, and use the following task orders:

(i) QuAC→ TrWeb → TrWik → SQuAD
(ii) SQuAD → TrWik → QuAC→ TrWeb
(iii) TrWeb → TrWik → SQuAD → QuAC
(iv) TrWik → QuAC→ TrWeb → SQuAD

**Model and Memory Details.** We use a pretrained BERT model augmented with an episodic memory to perform sparse experience replay. For text classification, we use the `[CLS]` token and a classifier to predict the class. For question answering, we apply two linear transformations to the BERT outputs for each token to predict the probability that the token is the start/end position of an answer. We implement the model using the huggingface library (Wolf et al., 2020). To train the model for both text classification and question answering, we use the Adam optimizer with a learning rate of $3e^{-5}$ and a training batch of size 32. We use the BERT base version and its default vocabulary in our experiments.

| Approach | Runtime |
|---|---|
| N. Random | 45m |
| Reservoir | 49m |
| Ring Buffer | 51m |
| Surprise | 1h 27m |
| Min. Margin | 1h 20m |
| Max. Loss | 46m |
| MoF | 2h 16m |

Table 2: Training time comparison of all seven memory population approaches for text classification, based on running task order (i) with one random seed on an NVIDIA GeForce RTX 3090.

The episodic memory is a buffer that stores veridical inputs and labels using the memory population methods mentioned above. We use an experience replay rate of 1% and memory capacity of 10%, which d'Autume et al. (2019) showed to be enough for good results (see Section 6 for additional experiments with varying memory sizes). We determine the memory capacity percentage based on the total size of the datasets. The retrieval process is performed randomly from the memory with a uniform probability. Regarding population for question answering task, all methods based on the number of classes were adapted to work based on the number of tasks. This is because question answering is a span prediction task with no classes.

## 6 Results

**Performance.** Text classification and question answering results are shown in Table 1, in the upper and lower sections respectively. For text classification, on average, *Reservoir* proved to be the best performing approach, with the *Naive Random* memory placing second. Overall, the standard deviations tend to have larger values than the differ-

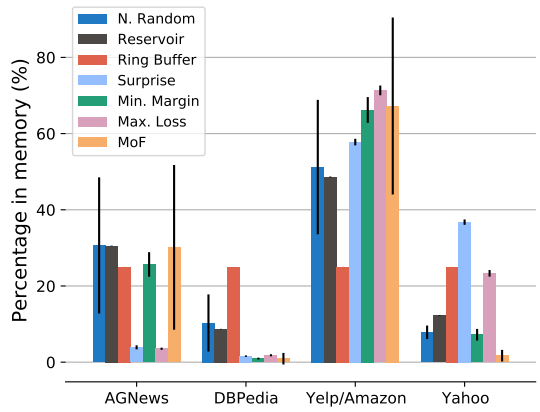


Figure 1: Percentage of samples in memory per task after training the model for text classification. Each color represents a different population method.

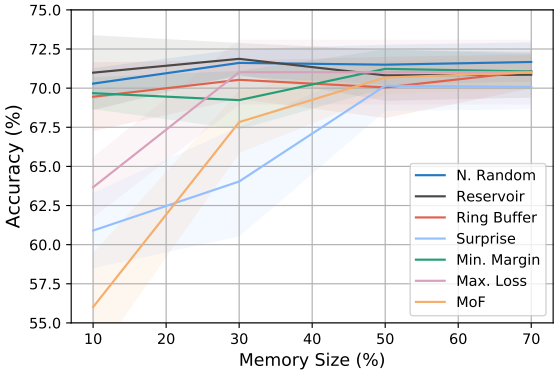


Figure 2: Sparse replay model performance for each population method with 10% to 70% memory size.

ences across approaches in many cases.

For the question answering problem, *Ring Buffer* memory performed best. Next, the *Naive Random*, *Reservoir*, *Surprise* and *Min. Margin* methods performed similarly. Compared to the text classification results, the differences in average performance across models and the standard deviations are substantially smaller. This difference could be due to the more homogeneous nature of the question answering tasks (i.e., start and end span predictions), contrary to the heterogeneous set of classes used in a stream of text classification tasks.

Overall, the *Max. Loss* and *Surprise* method results in lower returns, which is inconsistent with previous findings from CV (Hayes and Kanan, 2021; Isele and Cosgun, 2018). For the *MoF* approach, we were not able to replicate the improvement in performance (Chaudhry et al., 2019) in this NLP-specific application. We suspect that this is caused by the unsuitability of the `[CLS]` token for semantic similarity purposes (Reimers and Gurevych, 2019). Finally, *Reservoir* leads to the best results as it maintains a random sample over a global distribution that is not known in advance. This supports previous work on CV (Chaudhry et al., 2019), which defaults to the reservoir sampling due to its simplicity and efficiency.

We were able to confirm that the *Reservoir* and *Naive Random* methods are indeed the most efficient in terms of their required training time, together with *Max. Loss* and *Ring Buffer* (see Table 2). Notably, *MoF* is the most inefficient of the presented approaches, likely due to frequent updates of the average feature vector.

**Resulting Memory Composition.** Figure 1 depicts the resulting memory composition after training the model for text classification tasks. Specifically, it shows the percentage of items in memory per task normalized by the number of classes for all population methods. We join the Yelp and Amazon datasets because of their shared classes, resulting in an overpopulation in memory. As expected, *Ring Buffer* results in a balanced number of samples. Regarding the best performing methods, *Naive Random* and *Reservoir*, we observe similar behaviors, possibly explaining their similar performance. However, *Reservoir* better balances the number of instances per task, limiting the high number of examples stored for Yelp/Amazon.

Furthermore, certain methods result in an extremely imbalanced memory composition, which tends to hurt performance (Chrysakis and Moens, 2020). For instance, *Surprise* and *Max. Loss* are biased towards the last seen tasks (as they produce high surprise or loss), reducing the population of initial ones. Also, *MoF* stores nearby items, limiting the storage of previously unseen task instances.

**Memory Size Impact.** Figure 2 shows the performance for text classification for memory sizes between 10% and 70%. Most methods do not result in a performance advantage when the memory size increases, and between 50% and 70% capacity, all approaches tend to perform similarly.

However, methods with an extremely imbalanced memory composition, namely *Surprise*, *Max. Loss* and *MoF* (see Figure 1), benefit from higher memory capacities. Larger memory helps to avoid overwriting elements of past tasks, which counteracts imbalances in the composition of the memory.

**Forgetting and Memory Usage.** To better understand why some methods perform worse, we compare the model forgetting and memory usage of text classification task - order (ii). Forgetting is

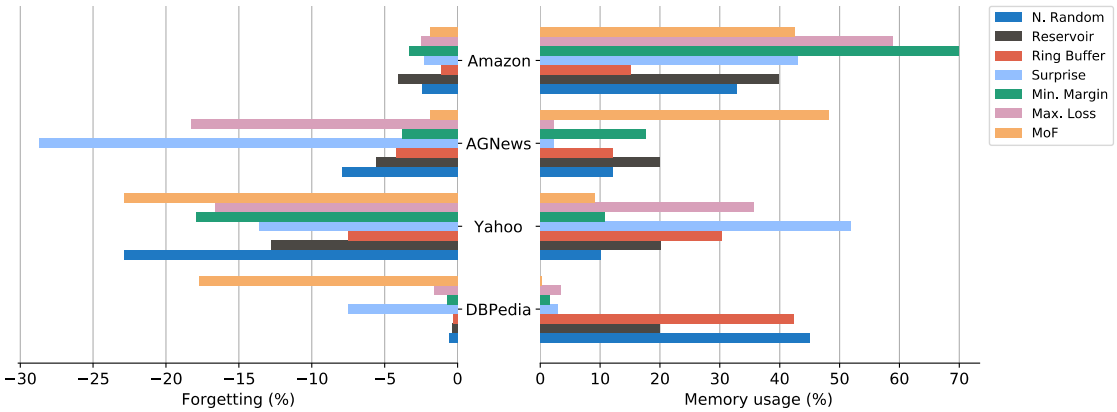


Figure 3: Double bar graph contrasting the percentages of forgetting and memory usage per task for all the population methods. Forgetting is computed by the difference between the current and previous model performance.

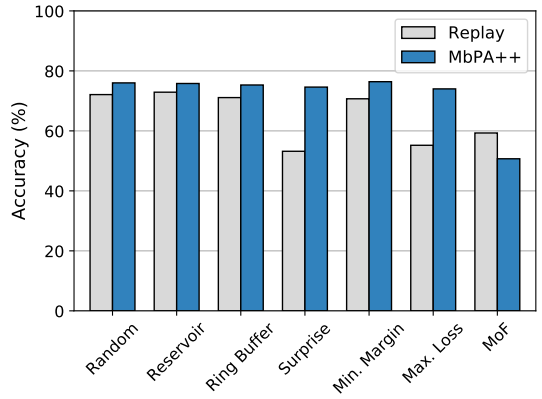


Figure 4: Influence of memory population methods when performing local adaptation to the replay model.

the difference between a task's final performance and the initial performance. Memory usage is the percentage of items in memory (non-normalized) belonging to a task.

Figure 3 shows a direct relationship between a high forgetting percentage and few elements in memory. This is the main reason why the *Surprise*, *Max. Loss* and *MoF* obtain the worst performance at 10% memory. However, there are some exceptions. *Surprise* and *Max. Loss* have many elements of the *Yahoo* dataset, but forgetting is also high. We hypothesize those methods store examples that are not representative of the task's global distribution, resulting in a possible underfitting of the model.

Interestingly, Figure 3 shows that *Reservoir* balances the number of samples in terms of tasks, which may be why this method surpass all others. Meanwhile, *Ring Buffer* gets lower performance by balancing memory in terms of classes (Figure 1), suggesting it is not the ideal way to fill the memory.

**Influence of Resulting Memory on Local Adaptation** As mentioned in Section 3, d'Autume et al. (2019) proposed the MbPA++ model, which is a replay model with an additional local adaptation step during inference. We analyze how the resulting memory influences the local adaptation process of the text classification tasks - order (ii).

Figure 4 shows that the resulting memories of *Surprise* and *Max. Loss* methods benefit from local adaptation. We hypothesize that this is due to the criteria of these methods. Intuitively, the memory samples hard examples, which might be beneficial for local adaptation but not for replay, potentially leading to overall poor performance. Relative to the other methods, there is no significant increase in performance by applying local adaptation. This could be because the model has already reached the upper bound performance. Lastly, *MoF* suffers from local adaptation, likely due to its suboptimal representations derived from `[CLS]` tokens.

## 7 Conclusion

In this work, we studied memory population methods for episodic memory in the context of lifelong language learning. Our empirical analysis shows that simple methods such as Naive Random and Reservoir are the best choice for text classification and question answering because they randomly sample the global distribution. However, in the case of question answering, a balanced memory in terms of tasks leads to better results.

## Acknowledgements

This work was supported by the European Research Council Advanced Grant 788506, the National Center for Artificial Intelligence CENIA FB210017 - Basal ANID, and Vicerrectoría de Investigación de la Pontificia Universidad Católica de Chile - Concurso Puente 2021.

## References


Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. 2019. *Gradient Based Sample Selection for Online Continual Learning*. Curran Associates Inc., Red Hook, NY, USA.

Vladimir Araujo, Julio Hurtado, Alvaro Soto, and Marie-Francine Moens. 2022. Entropy-based stability-plasticity for lifelong learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3721–3728.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Computer Vision – ECCV 2018*, pages 556–572, Cham. Springer International Publishing.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. 2019. On Tiny Episodic Memories in Continual Learning. *arXiv:1902.10486 [cs, stat]*.

Sen Cheng and Loren M. Frank. 2008. New experiences enhance coordinated neural activity in the hippocampus. *Neuron*, 57(2):303–313.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Aristotelis Chrysakis and Marie-Francine Moens. 2020. Online continual learning from imbalanced data. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1952–1961. PMLR.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Xu Han, Yi Dai, Tianyu Gao, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. Continual relation learning via episodic memory activation and reconsolidation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6429–6440, Online. Association for Computational Linguistics.

Tyler L. Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. 2020. Remind your neural network to prevent catastrophic forgetting. In *Computer Vision – ECCV 2020*, pages 466–483, Cham. Springer International Publishing.

Tyler L. Hayes and Christopher Kanan. 2021. Selective Replay Enhances Learning in Online Continual Analogical Reasoning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3497–3507.

Tyler L. Hayes, Giri P. Krishnan, Maxim Bazhenov, Hava T. Siegelmann, Terrence J. Sejnowski, and Christopher Kanan. 2021. Replay in Deep Learning: Current Approaches and Missing Biological Elements. *Neural Computation*, 33(11):2908–2950.

Nithin Holla, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020. Meta-learning with sparse experience replay for lifelong language learning. *arXiv preprint arXiv:2009.04891*.

David Isele and Akansel Cosgun. 2018. Selective experience replay for lifelong learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

David Lopez-Paz and Marc' Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Roger Ratcliff. 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. iCaRL: Incremental Classifier and Representation Learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542. IEEE.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Pablo Sprechmann, Siddhant Jayakumar, Jack Rae, Alexander Pritzel, Adria Puigdomenech Badia, Benigno Uria, Oriol Vinyals, Demis Hassabis, Razvan Pascanu, and Charles Blundell. 2018. Memory-based parameter adaptation. In *International Conference on Learning Representations*.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020a. LAMOL: LAnguage MOdeling for Lifelong Language Learning. In *International Conference on Learning Representations*.

Jingyuan Sun, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2020b. Distill and replay for continual language learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3569–3579, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shengyang Sun, Daniele Calandriello, Huiyi Hu, Ang Li, and Michalis Titsias. 2022. Information-theoretic online memory selection for continual learning. In *International Conference on Learning Representations*.

Jeffrey S. Vitter. 1985. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57.

Zirui Wang, Sanket Vaibhav Mehta, Barnabas Poczos, and Jaime Carbonell. 2020. Efficient meta lifelong-learning with limited memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 535–548, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu. 2019. Large scale incremental learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 374–382, Los Alamitos, CA, USA. IEEE Computer Society.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.