Momentum-based Weight Interpolation of Strong Zero-Shot Models for Continual Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large pre-trained, zero-shot capable models have shown considerable success both for standard transfer and adaptation tasks, with particular robustness towards distribution shifts. In addition, subsequent fine-tuning can considerably improve performance on a selected downstream task. However, through naive fine-tuning, these zero-shot models lose their generalizability and robustness towards distribution shifts. This is a particular problem for tasks such as Continual Learning (CL), where continuous adaptation has to be performed as new task distributions are introduced sequentially. In this work, we showcase that where fine-tuning falls short to adapt such zero-shot capable models, simple momentum-based weight interpolation can provide consistent improvements for CL tasks in both memory-free and memory-based settings. In particular, we find improvements of over $+4\%$ on standard CL benchmarks, while reducing the error to the upper limit of jointly training on all tasks at once in parts by more than half, allowing the continual learner to inch closer to the joint training limits.

1 Introduction

Continual Learning (CL) tackles the problem of learning from a non-stationary data stream, where training data is presented to the model not at once, but only in a sequence, and with limited capacity for retention and retraining. Not only does this require effective use of previously seen data, but also adaptation to novel context under continuously changing distribution shifts without catastrophic forgetting [11, 33, 34, 35]. Use cases are widespread, ranging from particularly compute-, time- or memory-limited to privacy-concerned applications [14, 3, 15, 39, 24]. Consequentially, previous research has introduced a wide range of methods to address training under continual shifts, such as through the use of efficient data replay [7, 3, 1, 27], regularization on the training dynamics [11, 35] or optimization procedures seeking for flat minima [22, 37]. Generally, these methods start from an untrained model which is then adapted to the data stream at hand.

While this has found practical success, more recently the use of large-scale pre-trained models ("foundation models" [2, 24]) has become ubiquitous, as they have shown strong zero-shot generalizability to a variety of downstream tasks, with strong robustness to distribution shifts [2]. Their application to the CL problem set, which tackles a continuous distribution shift, stands to reason, with recent works showing notable benefits in the use of foundation models [39, 20, 30, 41], particularly highlighting a reduction in catastrophic forgetting. Still, as learners are adapted to continuously shifting training distribution, even foundation models will suffer from forgetting through fine-tuning [40]. To maximize the benefits we can extract from the main continual learning process as well as the ability to classify novel samples at test time, it is thus important to minimize the impact on the generalizability of the adapted foundation model in order to account for potential further adaptations.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

To allow for improved deployment in the CL setting, in this work we show how momentum-based weight-interpolation can help remedy issues of such models adapted in a continual fashion. In particular, as we want to maximally retain the generalizability of our adapted foundation model, we introduce a bifurcated adaptation mechanism by retaining an additional copy of the initial foundation model (denoted as *slow* model). This slow model is excluded from the direct CL optimization process, and is only updated through linear momentum-interpolation with a task-adapted model copy (denoted as *fast* model). This is motivated by insights made in [40], who show that simple linear interpolation in weight space between the original zero-shot model and a variant fine-tuned to a task at hand allows for adaptation, while retaining better generalizability as compared to sole fine-tuning. However, retaining a large collection of fine-tuned task expert models in the CL setting is memory intensive, impractical, and undesired. Instead, we show that we can simulate the empirical benefits highlighted in [40] through repeated momentum interpolation between our foundation model and a continuously fine-tuned variant. This allows us to avoid the drawbacks of pure fine-tuning, while both specializing on the new stream of tasks, and retaining the generalizability of our foundation model.

Indeed, experiments on three standard CL benchmarks (Seq-CIFAR-10, Seq-CIFAR-100 and Seq-Tiny-ImageNet) show improvements in class- and task-incremental settings on both memory-based and memory-free methods by up to $+4\%$, and partly more than halving the error to the joint training performance bound. These results indicate that for practical usage of foundation models in a continuously distribution-shifting training scenario, momentum-based weight interpolation can be a reliable tool for consistent improvements that works well alongside any CL method.

2 Related Work

Regularization-based methods augment the training objective to mitigate forgetting by keeping the current parameters close to previous task parameters. Elastic Weight Consolidation (EWC) [12] performs Laplace approximations on the parameter posterior for each preceding task, using the means and covariances to regularize the current parameters via Mahalanobis distance minimization. Online Elastic Weight Consolidation (oEWC)[35] computes a momentum average of a single covariance matrix, and keeps the parameters from the last task only. Learning without Forgetting (LwF)[16] also keeps the parameters from the last task and adds a cross-entropy term between logits computed with the old and current parameters, using data from the current task. [21] show that dropout forces the model to learn a gating such that for different tasks, different paths of the network are active.
Rehearsal-based methods utilize Experience Replay [31] [32] by storing a small subset of the training data into a buffer, and continually replaying it as the model moves on to learn new tasks. Dark Experience Replay (DER) [4] introduces regularization in the rehearsal scheme by matching the logits of the past with the logits computed by the current network parameters. Gradient Episodic Memory (GEM) [17] and Average Gradient Episodic Memory (A-GEM) [8] enforce optimization constraints in the current task using data from past tasks. GDumb [28] greedily stores samples in memory, and only trains the model at test time using buffer data. DualNet [26] uses a slow network for learning task-agnostic features through Self-Supervised Learning, and a fast network for learning task-specific features. Contrastive Continual Learning (Co2L) [5] learns contrastive task-agnostic features, and trains a linear classifier using only buffer data.
Flatness-seeking methods aim to operate in flat minima regions for each task in sequence, thereby retaining antecedent performance. Finding Flat Minima (F2M) [38] independently adds small random noise to the parameters, thereby obtaining similar but different loss functions which are optimized jointly in order to locate flat minima. [23] studies how batch size, dropout and learning rate decay affect the model's ability to find flat basins. [19] uses the Sharpness-Aware Minimization (SAM) [9] procedure, which explicitly optimizes for parameters lying in flat basins.

3 Method

In CL, a model f_θ is trained on a sequence of T tasks, where for each task $t \in \{1, ..., T\}$ the learner only gets access to a subset of samples $D_t = \{(x_i, y_i)\}_{i=1}^{N_t}$, but is eventually evaluated on joint performance, i.e. we optimize $\theta^* = \arg\min_\theta \sum_{t=1}^{T} \mathbb{E}_{(x,y)\sim D_t} \left[L(f_\theta(x), y)\right]$. The main challenge is that at task t, the model has no access to data from previous tasks $\tilde{t} \in \{1, ..., t-1\}$, therefore violating the typical IID data assumption. In this work, we investigate both the class-incremental

setting, where subsets of classes are introduced in sequence, and the much easier task-incremental setting which jointly also provides respective task ids.

3.1 Momentum-based Weight Interpolation for Continual Learning (MCL)

To allow for effective and continuous adaptation of foundation models, we introduce momentum-based weight interpolation for CL. As our primary target is the retention of the generalizability and shift robustness of the underlying foundation model, it is important that minimal adaptation and fine-tuning is performed, while still allowing for a certain degree of adaptation to the target tasks at hand. For that, we suggest a retention of a *slow* model copy θ_{slow} which is kept disconnected from the entire adaptation process, while a second instantiation θ_{fast} is updated throughout the continual learning process. As θ_{fast} adapts to the target distribution at hand, at every iteration we simultaneously perform an iterative updating on our slow weights through weight-space interpolation:

$$\theta_{\text{slow}} = \tau \cdot \theta_{\text{slow}} + (1 - \tau) \cdot \theta_{\text{fast}}$$

where τ is our *momentum* hyperparameter. A simplified version of the procedure is summarized in Algorithm 1. As this mechanism is task- and memory-agnostic with no dependence on task boundaries, it can be applied to any continual learning framework, both memory-based and memory-free. And while straightforward and simple, the intuitively better retention of foundation model weights in the continual learning setting is well motivated. Beyond a conceptual connection to the Complimentary Learning Systems (CLS) [18, 6] theory from neuroscience which depicts human continual learning as an interplay of a fast adaptive and a slow retentive system, on a methodological level [10] show that maintaining a running average of weights leads to wider optima and retained generalization during the standard fine-tuning process of a pre-trained model. In addition, [40] showcase that zero-shot and fine-tuned model weights are often connected by a linear path which retains performance. It therefore stands to reason that our linear momentum-based interpolation across task iterations allows us to connect to the performance of our task-adapted fast variant, while maintaining the generalizability our foundation model weights θ_{slow}. The consequently sustained implicit optimization for a flatter minimum around θ_{slow}, which is only updated through momentum-based interpolation, has strong ties to improved generalization across task sequences in continual learning [38, 23, 9], which we see reflected in our benchmark experiments in the next section.

4 Experiments

Datasets. We evaluate our method on three datasets commonly used in the literature: CIFAR-10 [13], CIFAR-100 [13], and Tiny ImageNet. We split each dataset into several tasks of non-overlapping classes: Seq-CIFAR-10 consisting of 5 tasks (2 classes each) and Seq-CIFAR-100/Seq-Tiny-ImageNet consisting of 10 tasks (10 and 20 classes each, respectively).

Training. For our zero-shot model we use a pre-trained CLIP ViT-B/16 [29]. We built our CL experiments on [4] which implements several CL benchmarks in PyTorch [25]. All methods follow a standardized training protocol - trained on Nvidia 2080Ti's using SGD [36], a fixed learning rate and no scheduler, with the same fine-tuning budget of 10 epochs. We perform grid searches on a random train subset to select the best learning rate $\alpha \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$ as well as the best momentum strength $\tau \in \{0.995, 0.997, 0.999, 0.9995, 0.9997, 0.9999\}$. We refer the reader to the appendix (§A.1) for an ablation study of the hyperparameters.

Evaluation. For both Task Incremental Learning (Task-IL) and Class Incremental Learning (Class-IL) scenarios, we report the final classification accuracy over all encountered classes, with task identities also provided in the Task-IL setting (making it a noticeably easier problem to solve).

Table 1: Baselines

Baseline	CIFAR-10	CIFAR-100	Tiny-ImageNet
ZERO-SHOT	88.77	63.11	58.53
JOINT	97.53 ± 0.08	87.22 ± 0.54	78.86 ± 1.38

4.1 Experimental Results

In this section, we experiment with the use of momentum-based weight interpolation in three standard CL method categories: fine-tuning (pure SGD [36]), regularization-based (oEWC [35]), and rehearsal-based (DER++ [4] with buffer size 500 and 5000). The results presented below are obtained over three

Table 2: Continual Learning setting – training and evaluating on sequences of tasks.

Method	Momentum	Seq-CIFAR-10		Seq-CIFAR-100		Seq-Tiny-ImageNet	
		Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL
SGD	no	91.38 ± 0.04	98.17 ± 0.01	74.36 ± 0.03	93.59 ± 0.04	67.30 ± 0.08	82.12 ± 0.07
	yes	92.46 ± 0.11	98.43 ± 0.01	77.52 ± 0.37	94.98 ± 0.17	71.09 ± 0.28	85.22 ± 0.32
oEWC	no	90.67 ± 0.01	98.17 ± 0.01	74.07 ± 0.20	93.80 ± 0.02	66.60 ± 0.02	81.79 ± 0.02
	yes	91.87 ± 0.57	98.88 ± 0.12	77.25 ± 0.31	95.09 ± 0.01	71.57 ± 0.05	85.94 ± 0.07
DER++ (500)	no	94.65 ± 0.16	99.38 ± 0.10	76.68 ± 0.23	95.05 ± 0.09	71.05 ± 0.12	84.42 ± 0.22
	yes	95.73 ± 0.21	99.50 ± 0.04	82.01 ± 0.31	96.69 ± 0.03	75.11 ± 0.02	87.80 ± 0.27
DER++ (5000)	no	97.08 ± 0.04	99.60 ± 0.01	83.16 ± 0.20	97.03 ± 0.11	76.54 ± 0.10	88.44 ± 0.04
	yes	97.21 ± 0.11	99.62 ± 0.01	84.94 ± 0.07	97.13 ± 0.05	78.26 ± 0.14	89.00 ± 0.11

seeds, alongside which we provide the zero-shot lower bound (Tab. 1). Interestingly, the non-adapted zero-shot performance already in parts vastly outperforms comparable adaptation with state-of-the-art methods not relying on foundation models, with e.g. DER++ [4] reporting $72.70 \pm 1.36\%$ with a buffer of 500, and $85.40 \pm 0.49\%$ with a buffer of 5000 on CIFAR-10, while zero-shot performance of our foundation model already achieves 88.77%. This difference is even further exacerbated on Tiny-ImageNet, with $19.38 \pm 1.41\%$ and 39.02 ± 0.97 for buffer sizes of 500 and 5000 respectively, versus 58.53% for zero-shot performance, verifying the potential [20, 30] of foundation models in CL. To provide an upper bound, we train on all tasks jointly (Tab. 1). Since joint training is evaluated without task boundaries, this upper bound does not hold for Task-IL scenarios. Next, in Tab. 2 we present the results on the CL benchmarks. We empirically show that, as motivated in Sec. 3, keeping a momentum-interpolated version of the foundation model results in consistent improvements.

In particular, our results show that adaptation to the task distribution at hand is beneficial even with simple fine-tuning. Even when accounting for a change in learning rate (as noted in §4 and done for every baseline), we find that additional momentum-based weight interpolation offers consistent benefits in both class- and task-incremental settings, with nearly $+4\%$ improvement on both Seq-CIFAR-100 and Seq-Tiny-ImageNet. Furthermore, through momentum-updating, we can push simple fine-tuning close or even over the performance of a state-of-the-art CL framework (DER++). Additionally, we observe similar performance improvements even when applied on top of separate CL frameworks, both memory-free (oEWC, e.g. $74.07 \pm 0.20 \rightarrow 77.25 \pm 0.31$ on Seq-CIFAR-100) and memory-based (DER++ with 500 memory samples, $76.78 \pm 0.23 \rightarrow 82.01 \pm 0.31$). Interestingly, a momentum-extended DER++ with a buffer size of 500 also almost closes the gap in performance to the non-momentum based DER++ with a much larger buffer size 5000, which, even with such a large memory, also sees significant improvements on the particularly more complex CL tasks (Seq-Tiny-ImageNet, $76.54 \pm 0.10 \rightarrow 78.26 \pm 0.14$). This demonstrates that the need for buffer sizes in CL frameworks built around foundation models can decrease significantly (in this case, 10-fold) through momentum-based weight-space interpolation. We do note that while not necessary for the benchmarks at hand, longer task sequence may benefit from a re-synchronization of θ_{slow} and θ_{fast}.

Finally, we find that momentum-based DER++ with a buffer of 5000 even further closes the gap to the joint optimization upper bound - looking at the error, we find a drop of $0.45\% \rightarrow 0.32\%$ on Seq-CIFAR-10, $4.06\% \rightarrow 2.28\%$ on Seq-CIFAR-100, and $2.32\% \rightarrow 0.6\%$ on Seq-Tiny-ImageNet, which marks a nearly 75% reduction. Conclusively, these results indicate the significant benefits of retaining a momentum-updated model copy when introducing foundation models into the CL setting, both for consistent relative improvements, but also to minimize the performance drop when moving from the standard joint optimization to a continual learning scenario.

5 Conclusion

This work tackles the adaptation of large-scale pre-trained zero-shot models to continual learning (CL). To retain the strong generalizability and robustness of these models even under continuous fine-tuning, we propose the use of a momentum-based interpolation between a slow-moving zero-shot model excluded from the direct CL process and a task-adapted fast variant. Through this simple extension, we find consistent improvements in performance across three standard CL benchmarks (Seq-CIFAR-10, Seq-CIFAR-100, Seq-Tiny-ImageNet) on both memory-based and memory-free approaches, of in parts more than $+4\%$. In addition, we find the distance between continual learning and joint task optimization performance in some cases to even be more than halved. Based on these insights, the generalizability of large-scale pre-trained zero-shot models, and the simplicity of the proposed setup, we believe the adoption of our approach to be of high practical interest.

References

[1] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8218–8227, June 2021.

[2] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2021.

[3] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15920–15930. Curran Associates, Inc., 2020.

[4] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and SIMONE CALDERARA. Dark experience for general continual learning: a strong, simple baseline. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15920–15930. Curran Associates, Inc., 2020.

[5] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9516–9525, October 2021.

[6] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9516–9525, October 2021.

[7] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019.

[8] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*, 2019.

[9] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

[10] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[11] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[12] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[14] Cecilia Lee and Aaron Lee. Clinical applications of continual learning machine learning. *The Lancet Digital Health*, 2:e279–e281, 06 2020.

[15] Timothée Lesort, Oleksiy Ostapenko, Diganta Misra, Md Rifat Arefin, Pau Rodríguez, Laurent Charlin, and Irina Rish. Scaling the number of tasks in continual learning, 2022.

[16] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

[17] David Lopez-Paz and Marc' Aurelio Ranzato. Gradient episodic memory for continual learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[18] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychol Rev*, 102(3):419–457, Jul 1995.

[19] Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. An empirical investigation of the role of pre-training in lifelong learning. *arXiv preprint arXiv:2112.09153*, 2021.

[20] Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. An empirical investigation of the role of pre-training in lifelong learning, 2022.

[21] Seyed Iman Mirzadeh, Mehrdad Farajtabar, and Hassan Ghasemzadeh. Dropout as an implicit gating mechanism for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[22] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7308–7320. Curran Associates, Inc., 2020.

[23] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7308–7320. Curran Associates, Inc., 2020.

[24] Oleksiy Ostapenko, Timothee Lesort, Pau Rodríguez, Md Rifat Arefin, Arthur Douillard, Irina Rish, and Laurent Charlin. Continual learning with foundation models: An empirical study of latent replay, 2022.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[26] Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16131–16144. Curran Associates, Inc., 2021.

[27] Ameya Prabhu, Philip Torr, and Puneet Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *The European Conference on Computer Vision (ECCV)*, August 2020.

[28] Ameya Prabhu, Philip Torr, and Puneet Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *The European Conference on Computer Vision (ECCV)*, August 2020.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[30] Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*, 2022.

[31] R. Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychol Rev*, 97(2):285–308, Apr 1990.

[32] ANTHONY ROBINS. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

[33] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016.

[34] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4528–4537. PMLR, 10–15 Jul 2018.

[35] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress and compress: A scalable framework for continual learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4528–4537. PMLR, 10–15 Jul 2018.

[36] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 71–79, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[37] Guangyuan SHI, Jiaxin Chen, Wenlong Zhang, Li-Ming Zhan, and Xiao-Ming Wu. Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[38] Guangyuan SHI, JIAXIN CHEN, Wenlong Zhang, Li-Ming Zhan, and Xiao-Ming Wu. Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6747–6761. Curran Associates, Inc., 2021.

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

[40] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7959–7971, June 2022.

[41] Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan-Fang Li, Guilin Qi, and Gholamreza Haffari. Pretrained language model in continual learning: A comparative study. In *International Conference on Learning Representations*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Included all necessary experimental details, and built our work on top of a public codebase.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

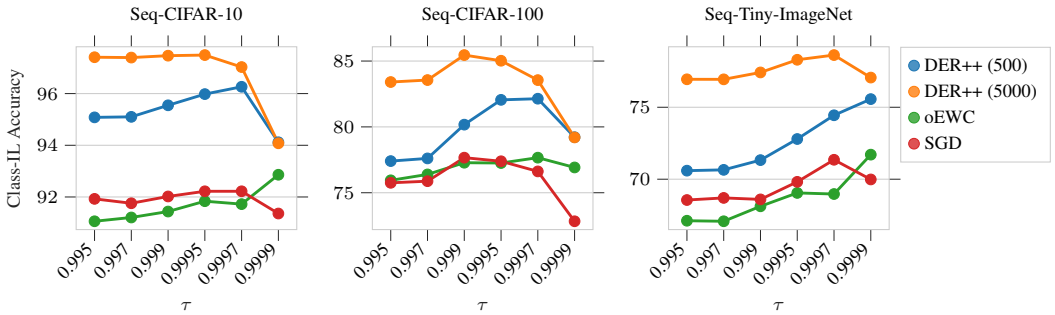

Figure 1: The effect of the hyper-parameter τ on the Class-IL Accuracy

A.1 Ablation study

Momentum strength. In Figure 1 we show how the momentum strength τ affects the model's Class-IL Accuracy. While we find that the optimal value of τ is dataset-dependent, it is encouraging that the vastly different methods show surprisingly similar behavior for a given dataset.

Restart frequency. Next, we examine whether it is beneficial to restart the fast weights θ_{fast} with the slow weights θ_{slow} during training (instead of only at the end as per default, i.e. Line 11 in Algorithm 1). To this end, we introduce a new hyperparameter *restart frequency* which specifies after how many gradient steps we perform a restart. From the results detailed in Figure 2, we find that restarting the fast weights is not beneficial to the generalization performance.

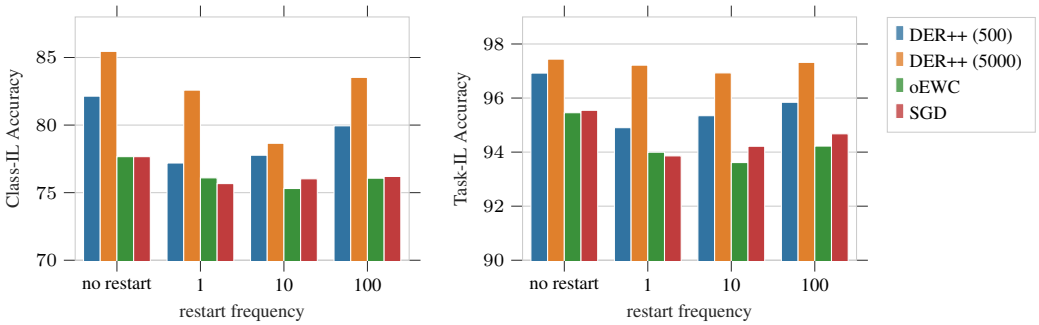

Figure 2: The effect of restarting the fast weights with the slow weights at various restart frequencies.

Update frequency. Finally, we examine whether it is beneficial to perform the update of the slow weights (Line 7 in Algorithm 1) at various frequencies. For this purpose, we introduce a new hyperparameter *update frequency* which specifies after how many gradient steps we update the slow weights. From the results summarized in Figure 3, we find that updating at frequencies higher than 1 (where 1 is the default behavior of our algorithm) does not provide a boost in performance.

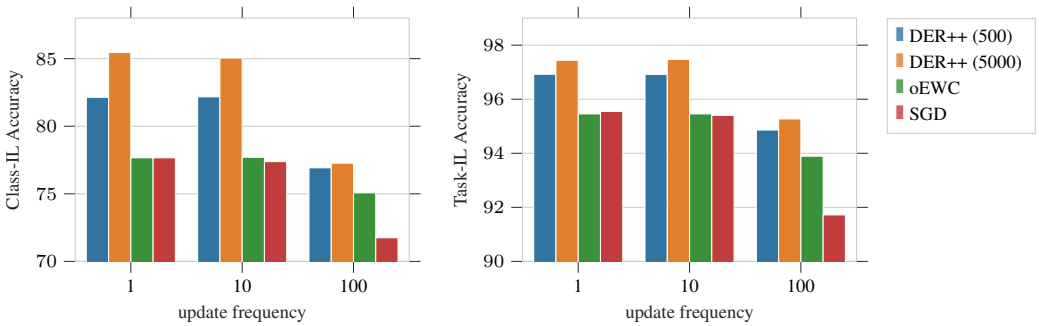

Figure 3: The effect of computing the momentum update at various update frequencies.

A.2 MCL Algorithm

Algorithm 1 Momentum-based Weight Interpolation for Continual Learning (MCL)

Require: Pre-trained weights θ_{pre}, Momentum $\tau \in [0, 1]$

$\theta_{\text{fast}} \leftarrow \theta_{pre}$
$\theta_{\text{slow}} \leftarrow \theta_{pre}$
for $t \leftarrow 1 \ldots$ num_tasks **do**
    for $e \leftarrow 1 \ldots$ num_epochs **do**
        for $(x, y) \sim D_t$ **do**
            $\theta_{\text{fast}} \leftarrow \theta_{\text{fast}} - \alpha \nabla \mathcal{L}(f_{\theta_{\text{fast}}}(x), y)$
            $\theta_{\text{slow}} \leftarrow \tau \cdot \theta_{\text{slow}} + (1 - \tau) \cdot \theta_{\text{fast}}$
        end for
    end for
end for
$\theta_{\text{fast}} \leftarrow \theta_{slow}$