Continual Pre-Training of Large Language Models: How to (re)warm your model?

Kshitij Gupta[*]
Université de Montréal
Mila

Benjamin Therien[*]
Université de Montréal
Mila

Adam Ibrahim[*]
Université de Montréal
Mila

Mats L. Richter
Université de Montréal
Mila

Quentin Anthony
Université de Montréal
EleutherAI

Eugene Belilovsky
Concordia University, Mila
Université de Montréal

Irina Rish
Université de Montréal
Mila

Timothée Lesort
Université de Montréal
Mila

Abstract

Large language models (LLMs) are routinely pre-trained on billions of tokens, only to restart the process over again once new data becomes available. A much cheaper and more efficient solution would be to enable the continual pre-training of these models, i.e. updating pre-trained models with new data instead of re-training them from scratch. However, the distribution shift induced by novel data typically results in degraded performance on past data. Taking a step towards efficient continual pre-training, in this work, we examine the effect of different warm-up strategies. Our hypothesis is that the learning rate must be re-increased to improve compute efficiency when training on a new dataset. We study the warmup phase of models pre-trained on the Pile (upstream data, 300B tokens) as we continue to pre-train on SlimPajama (downstream data, 297B tokens), following a linear warmup and cosine decay schedule. We conduct all experiments on the Pythia 410M language model architecture and evaluate performance through validation perplexity. We experiment with different pre-training checkpoints, various maximum learning rates, and various warmup lengths. Our results show that while rewarming models first increases the loss on upstream and downstream data, in the longer run it improves the downstream performance, outperforming models trained from scratch—even for a large downstream dataset.

1 Introduction

Large pre-trained models have enabled massive performance improvements for many downstream tasks in vision Kirillov et al. [2023], Oquab et al. [2023] and language [Brown et al., 2020, Zhao et al., 2023]. However, training these foundation models is prohibitively expensive. Existing works aim to reduce the cost of large-scale model development by enabling low-cost hyperparameter optimization Yang et al. [2022] or providing guidelines for maximizing performance under a given compute budget Hoffmann et al. [2022]. However, these works assume that models will be *trained from scratch*.

As the amount of data available for pre-training is ever-growing, new and improved datasets (e.g. RedPajama and SlimPajama Together.xyz [2023], Soboleva et al. [2023], Touvron et al. [2023]) will

---

[*]Equal contribution. Correspondence to {kshitij.gupta, benjamin.therien, ibrahima}@mila.quebec

continue to become available. Should practitioners always combine existing datasets (e.g. Pile Gao et al. [2020]) and *train from scratch* to obtain the best performance? Doing so would quickly become prohibitively expensive and fails to leverage existing pre-trained models.

Our approach circumvents the need for complete re-training by continuing to pre-train existing models on new data. We refer to this as "continual pre-training" and the goal is to minimize the loss on new data while maintaining low loss on previous data. Continual pre-training is a critical challenge since it can lead to catastrophic forgetting French [1999]. Moreover, the potential long sequence of training stages may make common continual learning techniques such as replay Rebuffi et al. [2017], Ostapenko et al. [2022] or regularisation Kirkpatrick et al. [2017], Farajtabar et al. [2020] not compute efficient enough Lesort et al. [2023]. A simple and – from a compute cost perspective – scalable solution to limit forgetting in such situations is to (only) progressively decrease the learning rate every time new data becomes available Mirzadeh et al. [2020], Winata et al. [2023]. However, this solution is limited because repeatedly decreasing the learning rate would cause it to eventually become too small if the number of training stages becomes high.

In this work, we take a step towards efficient continual pre-training by studying how to re-increase a small learning rate to keep training a pre-trained language model on new data. We refer to this as *re-warming* the model. Re-warming the model should improve learning efficiency by avoiding a vanishing learning rate. We study warm-up strategies on Pythia 410M model with various amounts of data, maximum learning rates and different pre-trained checkpoints. This would allow a model trained initially on a large dataset to benefit from resuming training on a newer large dataset without having to retrain from scratch. In order to simulate this setting, we fix our initial pre-training dataset to be Pile and the newer dataset to be SlimPajama. We hope that this may guide the adaptation of existing LLMs to future new datasets. Our results show that:

1. The length of the warmup phase does not appear to have a significant effect on the loss.
2. Adjusting the maximum learning rate can help trade-off between upstream and downstream performance; increasing the maximum learning rate leads to stronger adaptation to the downstream dataset (SlimPajama), while smaller learning rates preserve more performance on the upstream dataset (Pile).
3. Continual pre-training with the latest pre-trained checkpoint improves performance.

2 Setup

In our setup, the upstream (or pre-training) dataset is the Pile Gao et al. [2020]. The downstream (or fine-tuning) dataset is SlimPajama Soboleva et al. [2023]. SlimPajama is an extensively deduplicated version of RedPajama Together.xyz [2023] which is built based on the LLama dataset Touvron et al. [2023]. In this work, we use *fine-tuning* and downstream continual pre-training interchangeably. However, in our continual pre-training setting, we note that the downstream dataset is on the scale of the previous pre-training dataset (i.e. very large, unlike many fine-tuning datasets).

The SlimPajama dataset is built from similar sources as the Pile but with a higher quantity of data. Therefore, some upstream data may be repeated during downstream pre-training. Our experimental setup is comparable to the setup of Ash and Adams [2020], where they train a classifier on half of the samples of a dataset first, and fine-tune it later on all samples. They show that warm starting for image classification is challenging. Using a model pre-trained on the Pile and continuing the pre-training on SlimPajama, we follow an analogous setup for causal language modeling.

Datasets – We use the Pile with the same weights as Black et al. [2022] for validation. We shuffle and randomly sample the SlimPajama dataset Soboleva et al. [2023] to form the $\sim$297B token training dataset and $\sim$316M validation token dataset. We do not use replay. We use the same tokenizer as Black et al. [2022] that is trained specifically on the Pile.

Model – We use the 410M Pythia pre-trained on the Pile [Biderman et al., 2023], i.e. GPT-NeoX Black et al. [2022] models. We do not use flash attention [Dao et al., 2022].

3 Continual Warm-up

3.1 How long to warm up?

In the literature, warm-up is usually conducted on at most 1% of the data Zhao et al. [2023]. In this experiment, we investigate if the results are sensitive to this hyper-parameter.

Setup: We experiment with different warm-up lengths for a schedule of 297B tokens: 0%, 0.5%, 1%, and 2% of the data and measure the performance after the first 50B tokens. From a different perspective, we could see this experiment as running a 1% warm-up on different amounts of data. We hypothesize that warming up for a larger number of iterations could lead to a smoother transition with subsequent performance improvements.

Results: The results of this experiment are provided in Fig. 1. They show that the amount of data used for warming up the learning rate does not significantly influence the perplexity on the downstream task (learning) or the upstream task (forgetting). These results invalidate our hypothesis that using more tokens for warm-up can smooth the transition and show that linear warmup does not help in this setting. Indeed, the length of the warmup phase does not appear to have a significant effect on the Pile and SlimPajama validation losses. This is in spite of the model trained without any progressive warm up experiencing an initial *choatic phase* causing a spike in the loss in its first few iterations of training – this phenomenon is also referred to as stability gap [Lange et al., 2023, Caccia et al., 2022].
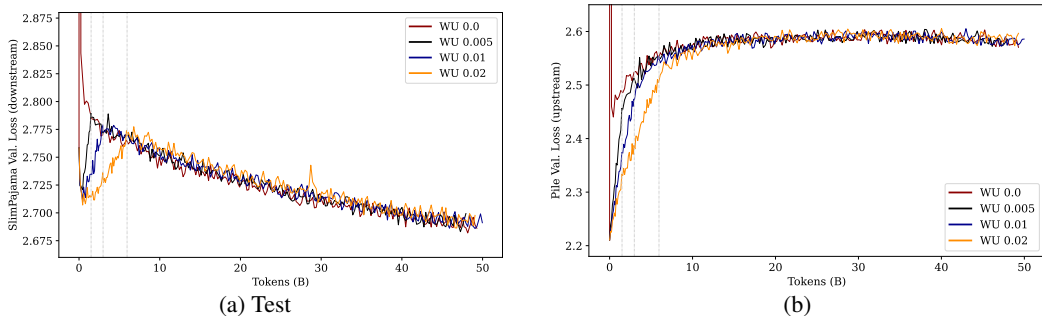

Figure 1: (*a*) Evolution of perplexity on SlimPajama while fine-tuning with various amounts of tokens for warm-up. (*b*) perplexity on the same experiments on the Pile validation set (upstream). $MaxLR = 3 \cdot 10^{-4}$, $MinLR = 0.1 \cdot MaxLR$. This figure shows that at that scale, the length of the warm-up phase does not significantly influence results.

3.2 How high to warm up?

One objective of re-warming the learning rate is to enable compute-efficient continual pre-training. A learning rate that is too small may lead to inefficient learning on the downstream dataset, whereas, a learning rate that is too large may lead to catastrophic forgetting of the upstream dataset. One important aspect of re-warming the learning rate is to decide how high to increase it. Therefore, in this experiment, we vary the maximum learning rate to assess its effect on performance.

Setup: We fix the length of the warm-up phase to the default amount of 1% of the training data and vary the maximum learning rate. We experiment with the default value of $3 \cdot 10^{-4}$ used for pre-training Pythia 410M Biderman et al. [2023], $1.5 \cdot 10^{-4}$, and $6 \cdot 10^{-4}$. For the post-warmup cosine decay phase, we set the final learning rate to 10% of the maximum learning rate. The learning rate schedule we used decays to the minimum learning rate at 240B tokens and is constant thereafter. The runs are reported to the end of 240B tokens (the end of decay period).

Results: The results of this experiment are provided in Fig. 2 (with an additional plot in the appendix Fig. 6). We observe, at the end of training, that larger maximum learning rates improve performance on downstream data, while they hurt performance on upstream data. Conversely, a smaller maximum learning rate improves performance on upstream data, while limiting adaptation to downstream data, causing decreased performance. These findings show that altering the maximum learning rate can be an effective way to trade-off between downstream and upstream performance. Additionally, we observe a general trend: fine-tuning on SlimPajama, causes the model to forget what has been learned on the Pile, leading to an increase in the Pile validation perplexity. Generally, rewarming then decaying the learning rate appears necessary to learn well on the downstream task. Moreover, while keeping a constant learning is initially advantageous on Pile, this advantage vanishes when training long enough on SlimPajama. Finally, a model trained from scratch only on SlimPajama performs worse on SlimPajama than models pretrained on Pile in spite of being optimised solely for the downstream task, highlighting positive transfer between the two datasets.

3.3 Comparing with from Scratch Training

In this experiment, we compare continually pretrained models with models trained from scratch.
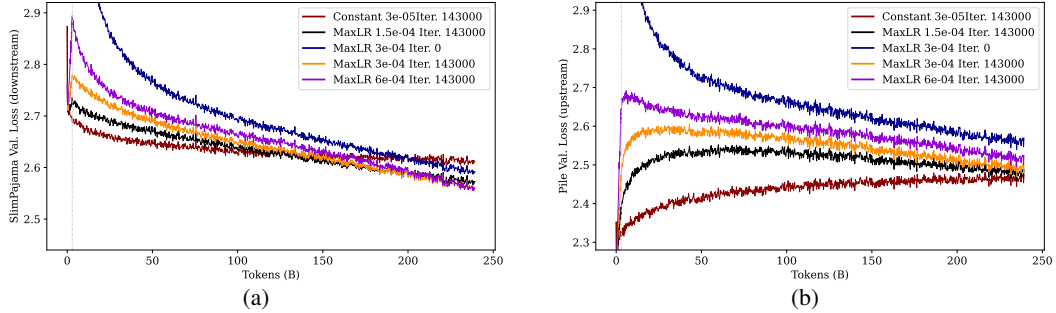
Figure 2: Evolution of loss on *(a)* SlimPajama and *(b)* Pile for different maximum learning rates. The blue curve reports a model trained from scratch. Growing the maximum learning rate consistently decreases the final loss on downstream data but increase it on the upstream data. At convergence, the models being continually pre-trained outperform the scratch and constant LR baselines on SlimPajama. However, the constant learning rate model achieves best performance within the first 100B tokens. The from-scratch baseline consistently improves its performance on Pile, while being trained on SlimPajama, showing the significant synergy between both datasets.

Setup: We train a model from random initialization using the same cosine decay schedule as the $MaxLR = 3 \cdot 10^{-4}$ model in 3.2.

Results: As we can see in Fig. 2, all the continually pretrained models with a warm-up perform better than the model trained from scratch. This shows that continually pretraining instead of retraining might improve performance even when the downstream dataset is on the scale of the upstream dataset and overlaps with the upstream dataset. We also observe that, after 200B tokens, the model trained from scratch performs better than the model continually pretrained using a constant learning rate.

3.4 Re-warming on the same data

In the previous experiments we saw that continually pretraining on new data leads to a quick increase of loss on past data, that decrease later. The increase is higher when the max learning rate is bigger. One hypothesis for the increase in loss is that the distribution shift between upstream and downstream data disturbs the training process. To assess this hypothesis, we apply our warm-up policy in a setting with no distribution shift. That is, we replicate our experiments from figures 2b and 6 by fine-tuning on Pile.

Setup: In this experiment, instead of fine-tuning on SlimPajama data, we fine-tune on 50B tokens of the Pile data with the same parametrization of the warm-up policy as Sec. 3.2 experiments.
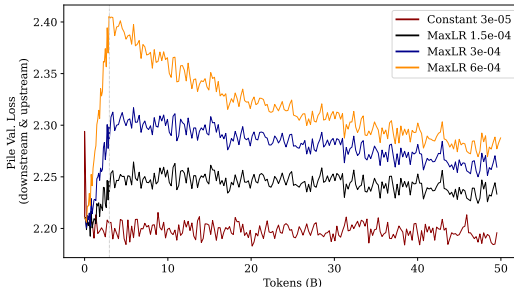

Figure 3: Pile validation loss while fine-tuning again on the Pile. Warm-up phenomenon observed in Sec. 3.2 is also observed applied to fine-tuning again on the same data distribution. Warm-up length=1% downstream tokens, $MinLR = 0.1 \cdot MaxLR$.

Results: Fig. 3, shows that re-warming the learning rate while continuing to pre-train on the Pile has a similar effect as re-warming on SlimPajama data Fig. 2b when looking at the downstream validation loss. This suggests that the distribution shift between Pile and SlimPajama is not solely to blame for the negative impact of re-warming the learning rate observed in sec. 3.2, and that the optimization dynamics also plays a role in this increase of loss.

4 Conclusion

Our experiments demonstrate that warming up to higher maximum learning rates helps models pre-trained on the Pile adapt to SlimPajama, while a smaller maximum learning rater preserves performance on the Pile. In both cases, however, models that are rewarmed can improve over models trained from scratch or without rewarming then redecaying the learning rate. These results motivate

the use of continual pre-training on new datasets rather than restarting training from scratch. More research is needed, however, to establish similar results for larger model scales, different distribution shifts, and verify that this strategy can be applied repeatedly to update models. We discuss limitations of this work in more details in C.

Acknowledgements

References

Jordan Ash and Ryan P Adams. On warm-starting neural network training. *Advances in neural information processing systems*, 33:3884–3894, 2020.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. *arXiv preprint arXiv:2304.01373*, 2023.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. Gpt-neox-20b: An open-source autoregressive language model, 2022.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 1877–1901, 2020. URL https://arxiv.org/abs/2005.14165.

Lucas Caccia, Jing Xu, Myle Ott, Marc'Aurelio Ranzato, and Ludovic Denoyer. On anytime learning at macroscale. *arXiv preprint arXiv:2106.09563*, 2021.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=N8MaByOzUfb.

Tuhin Chakrabarty, Christopher Hidey, and Kathy McKeown. IMHO fine-tuning improves claim detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 558–563, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1054. URL https://aclanthology.org/N19-1054.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16712–16721, 2022.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3762–3773. PMLR, 2020. URL https://arxiv.org/abs/1910.07104.

Enrico Fini, Victor G Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2022.

Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. ISSN 13646613. doi: 10.1016/S1364-6613(99)01294-2. URL `https://www.sciencedirect.com/science/article/abs/pii/S1364661399012942`.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, September 2021. URL `https://doi.org/10.5281/zenodo.5371628`.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020. URL `https://arxiv.org/abs/2004.10964`.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A Smith, and Luke Zettlemoyer. Demix layers: Disentangling domains for modular language modeling. *arXiv preprint arXiv:2108.05036*, 2021. URL `https://arxiv.org/abs/2108.05036`.

Rujun Han, Xiang Ren, and Nanyun Peng. ECONET: Effective continual pretraining of language models for event temporal reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5367–5380, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.436. URL `https://aclanthology.org/2021.emnlp-main.436`.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. URL `https://arxiv.org/abs/2203.15556`.

Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. The impact of non-stationarity on generalisation in deep reinforcement learning. *arXiv preprint arXiv:2006.05826*, 2020. URL `https://arxiv.org/abs/2006.05826.pdf`.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. Towards continual knowledge learning of language models. *arXiv preprint arXiv:2110.03215*, 2021. URL `https://arxiv.org/abs/2110.03215`.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. 2022.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. Lifelong pretraining: Continually adapting language models to emerging corpora. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 1–16, May 2022. doi: 10.18653/v1/2022.bigscience-1.1. URL `https://aclanthology.org/2022.bigscience-1.1`.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*, 2023a. URL `https://openreview.net/forum?id=m_GDIItaI3o`.

Zixuan Ke, Yijia Shao, Haowei Lin, Hu Xu, Lei Shu, and Bing Liu. Adapting a language model while preserving its general knowledge. *arXiv preprint arXiv:2301.08986*, 2023b.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proc. of the national academy of sciences*, 2017. URL `https://www.pnas.org/content/pnas/114/13/3521.full.pdf`.

Matthias De Lange, Gido M van de Ven, and Tinne Tuytelaars. Continual evaluation for lifelong learning: Identifying the stability gap. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=Zy350cRstc6`.

Timothee Lesort, Oleksiy Ostapenko, Pau Rodriguez, Md Rifat Arefin, Diganta Misra, Laurent Charlin, and Irina Rish. Challenging common assumptions about catastrophic forgetting. 2023.

Timothée Lesort, Hugo Caselles-Dupré, Michael Garcia-Ortiz, Jean-François Goudou, and David Filliat. Generative models from the perspective of continual learning. In *IJCNN - International Joint Conference on Neural Networks*, Budapest, Hungary, Jul 2019. URL `https://hal.archives-ouvertes.fr/hal-01951954`.

Timothée Lesort, Massimo Caccia, and Irina Rish. Understanding continual learning settings with data distribution drift analysis. *arXiv preprint arXiv:2104.01678*, 2021.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. URL `https://arxiv.org/abs/1711.05101`.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-demo.25. URL `https://aclanthology.org/2022.acl-demo.25`.

Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations*, 2021.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 33:7308–7320, 2020.

Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018. URL `https://arxiv.org/abs/1710.10628`.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

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.

Yujia Qin, Jiajie Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Elle: Efficient lifelong pre-training for emerging data. *arXiv preprint arXiv:2203.06311*, 2022. URL `https://arxiv.org/abs/2203.06311`.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021. URL `https://arxiv.org/abs/2112.11446`.

Dushyant Rao, Francesco Visin, Andrei A. Rusu, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Continual unsupervised representation learning. 2019. URL `https://arxiv.org/pdf/1910.14481.pdf`.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. URL `https://arxiv.org/abs/1611.07725`.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022. URL `https://arxiv.org/abs/2211.05100`.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. Fine-tuned language models are continual learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6107–6122, 2022.

Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. Continual learning in generative adversarial nets. *CoRR*, abs/1705.08395, 2017. URL `http://arxiv.org/abs/1705.08395`.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. `https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama`, 2023. URL `https://huggingface.co/datasets/cerebras/SlimPajama-627B`.

Together.xyz. Redpajama: An open source recipe to reproduce llama training dataset, 2023. URL `https://github.com/togethercomputer/RedPajama-Data`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. URL `https://arxiv.org/abs/2302.13971`.

Genta Indra Winata, Lingjue Xie, Karthik Radhakrishnan, Shijie Wu, Xisen Jin, Pengxiang Cheng, Mayank Kulkarni, and Daniel Preotiuc-Pietro. Overcoming catastrophic forgetting in massively multilingual continual learning. *arXiv preprint arXiv:2305.16252*, 2023.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*, 2019.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, David Farhi, Jakub Pachocki, Xiaodong Liu, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. In *NeurIPS 2021*, March 2022. URL `https://www.microsoft.com/en-us/research/publication/tuning-large-neural-networks-via-zero-shot-hyperparameter-transfer/`.

Mengyao Zhai, Lei Chen, Frederick Tung, Jiawei He, Megha Nawhal, and Greg Mori. Lifelong gan: Continual learning for conditional image generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2759–2768, 2019.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. URL `https://arxiv.org/abs/2303.18223`.

A Methodology

A.1 Dataset Composition

Table 1: Token counts and train data weights for our subsampled version of SlimPajama.

Dataset	Sampling %	Train	Val
StackExchange	2.0	9.95B	13.08M
Arxiv	2.5	13.77B	22.73M
Wikipedia	4.5	11.78B	15.79M
Book	4.5	14.22B	22.04M
Github	4.5	15.41B	22.42M
C4	15.0	78.49B	72.49M
Commoncrawl	67.0	153.25B	147.28M
Totals	100	296.86B	315.83M

A.2 Hyperparameters

Hyperparameters – We use the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 10^{-8}$, and a weight decay of 0.1. The maximum learning rate is varied in our experiments $\{1.5 \cdot 10^{-4}, 3 \cdot 10^{-4}, 6 \cdot 10^{-4}\}$. We use cosine learning rate decay to a minimum of $0.1 \cdot MaxLR$. All warmup lengths are calculated based on the full downstream dataset size (297B tokens). We note that our cosine decay schedule reaches the minimum learning rate at 240B tokens and is constant thereafter. We set gradient clipping to 1.0. Training is conducted at half-precision (FP16), without dropout.

B Related Work

Large Language Models: LLMs are usually trained with Adam (e.g., GPT3 [Brown et al., 2020], BLOOM [Scao et al., 2022], Gopher [Rae et al., 2021], Pythia Biderman et al. [2023]) or AdamW (e.g., Chinchilla [Hoffmann et al., 2022], LLaMA [Touvron et al., 2023]). In all the aforementioned models, the learning rate schedule consists of a warm-up followed by a cosine decay to 10% of the maximum learning rate.

Unsupervised Continual Learning: In this paper, we investigate various warm-up strategies for the continual pre-training of LLMs. Continual pre-training uses a similar type of training objectives as continual self-supervised training. Self-supervised pre-training was also studied in vision datasets for image generation Seff et al. [2017], Lesort et al. [2019], Zhai et al. [2019], Nguyen et al. [2018], Davari et al. [2022] or representation learning Fini et al. [2022], Madaan et al. [2021], Rao et al. [2019]. In language, continual pre-training was studied under the name of domain adaptation pre-training Ke et al. [2023a], Scialom et al. [2022], Gururangan et al. [2021], Qin et al. [2022] where the new dataset comes from a new domain. Another setting is where different datasets are generated at different points in time Han et al. [2021], Jin et al. [2022], Jang et al. [2021, 2022], Loureiro et al. [2022]. In our setup, the scenario is closer to domain adaptation pre-training, because we do not take into account the temporality of data.

Monitoring Learning Rate for Continual Training of Language Models: In continual learning (CL), models are trained on sequences of datasets. Therefore, the data is not *independent and identically distributed* which can lead the model to lose plasticity or forget. In such situations, particular monitoring of the learning rate schedule can be beneficial.

In CL of language models Caccia et al. [2021], Ke et al. [2023a], Loureiro et al. [2022], Han et al. [2021], Loshchilov and Hutter [2018], Scialom et al. [2022], Winata et al. [2023] different approaches have been evaluated: constant learning rate Ke et al. [2023a], Scialom et al. [2022], progressive decrease Winata et al. [2023] or warm-up then decrease Caccia et al. [2021].

However, to the best of our knowledge, no existing work studies specifically the influence of the warm-up phase in the context of continual pre-training for large language models.

C   Limitations

Data similarity and overlapping: In our experimental setup, upstream and downstream data have a high similarity, notably because of data overlap. Since in continual learning, different types of shifts can lead to variations in performance Lesort et al. [2021], our results may not generalize to setups with different distribution shifts, such as language domain adaptation pre-training setups Xu et al. [2019], Gururangan et al. [2020], Ke et al. [2023a], Chakrabarty et al. [2019], Ke et al. [2023b]. Nevertheless, comparing Fig. 6 and Fig. 7, we see that the results are not identical when fine-tuning on the Pile or when fine-tuning on SlimPajama. A possible explanation is that even a slight shift in data distribution can lead to a significant perturbation of the learning dynamics. For example, in the context of image classification, Igl et al. [2020] show how a sudden transition of 10 to 20 % of the labels in the dataset can have a significant impact on the downstream performance (see Fig. 5 of [Igl et al., 2020]).

Experiments Scale: As described in Sec. 2, our investigation explores models of size 410M and fine-tuning dataset of size 297B tokens. While this is a preliminary study, in future work, we plan to verify whether our conclusions hold at different model scales (e.g., 3B or 7B) and different dataset scales (e.g., 100B and 600B). Moreover, we plan to test our models throughout using benchmarks such as HELM [Liang et al., 2022] or Harness [Gao et al., 2021] instead of only loss or perplexity, as these benchmarks can provide important insight into the evolution of model capabilities.

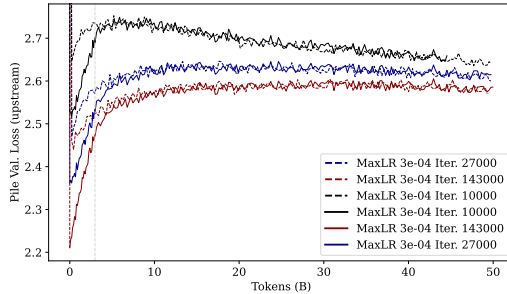## D   Upstream loss when fine-tuning various checkpoints.

Figure 4: Pile validation loss of models trained from the fully converged checkpoint, the upstream saturation point, and $1/2$ of the upstream saturation point. The experiments for this figure are described in Sec. F.
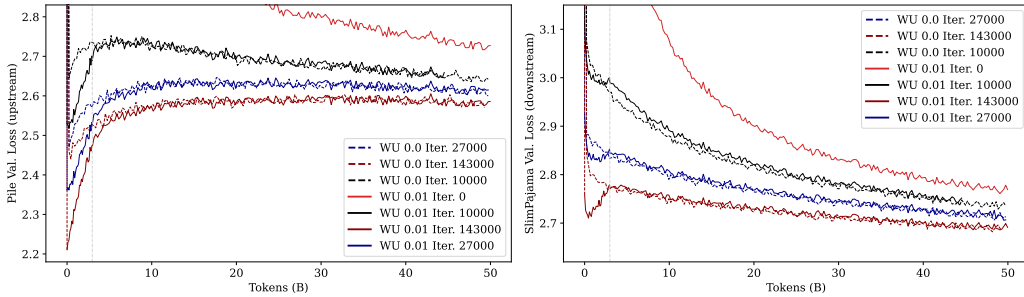

Figure 5: Training from a pre-trained checkpoint achieves lower Pile and SlimPajama validation loss faster than training from scratch.

E Perplexity on Pile vs SlimPajama

Fig. 7 shows that the training first increases perplexity on both the Pile and SlimPajama data but reduces after on both. Interestingly, Fig. 7 show a linear relationship between SlimPajama perplexity and the Pile perplexity when fine-tuning on the Pile, while it was not the case while fine-tuning on SlimPajama (Fig. 2b).
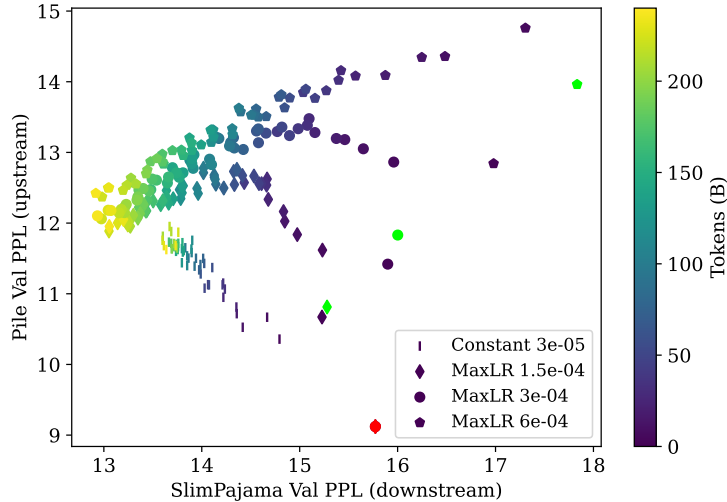

Figure 6: Perplexity downstream vs perplexity upstream, RP fine-tuning. Green points refer to the ends of the warm-up phases. The red point represents the perplexity before starting the downstream fine-tuning. Increasing the maximum learning rate improves performance on the downstream data, but causes forgetting on the upstream. This plot reports the same results as Fig. 2.
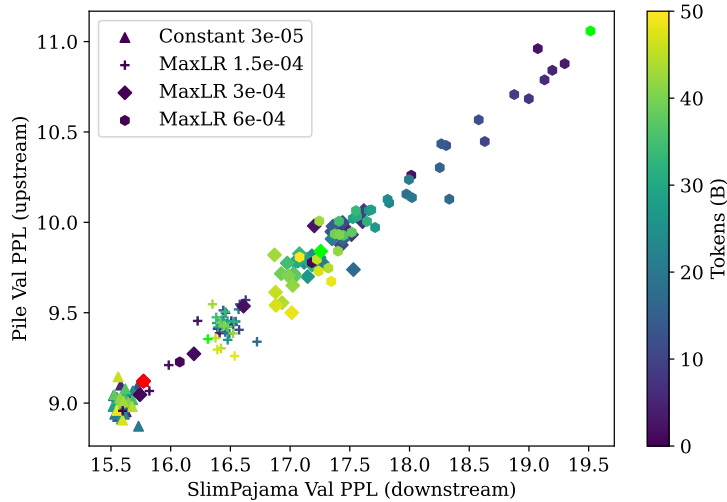

Figure 7: Perplexity on the Pile vs perplexity on SlimPajama when fine-tuning on the Pile with various maximum learning rates. Warm-up token=1% downstream tokens, $MinLR = 0.1 \cdot MaxLR$. Green points refer to the end of the warm-up phase.

F Evaluating Earlier Checkpoints

Setup: We select three checkpoints from model pre-training to test if warm-up strategies benefit from starting with non-converged checkpoints. Our hypothesis is that selecting checkpoints farther from

convergence may benefit adaptation to the downstream task as these checkpoints may be located at more favorable points in the loss landscape.

To select significantly different checkpoints, we compare the last pre-training checkpoint (i.e. Pythia 410M after $143,000$ iters), to an earlier checkpoint achieving a Pile validation loss near the maximum Pile validation loss attained by all models in Fig. 2 (bottom) (~ 2.5), and a third checkpoint in between the two other checkpoints.
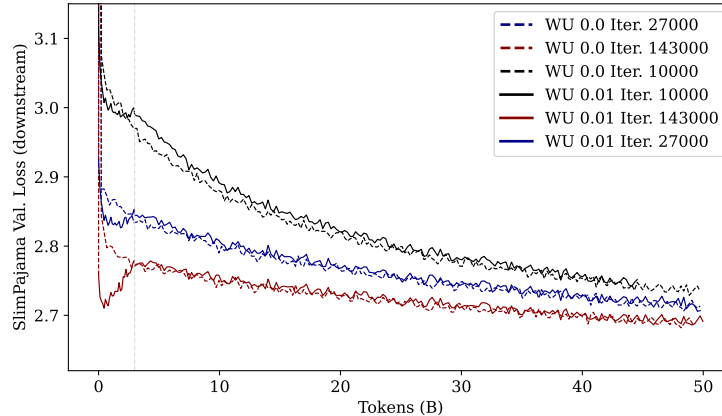

Figure 8: Pile validation loss of models trained from the fully converged checkpoint, the upstream saturation point, and $1/2$ of the upstream saturation point. Black colour designs for the earlier checkpoint, red colour the latest checkpoint and blue colour the in-between one.

Results: The evolution of the validation losses on SlimPajama are provided in Fig. 8 and the evolution of the validation losses on the Pile is provided in appendix D. We see in Fig. 8 that, in our setup, selecting earlier checkpoints for later fine-tuning does not lead to improvement in downstream performance. Therefore, selecting the latest checkpoint is the best option. We can conclude that the pre-training did not lead the model into a loss of plasticity that would make the model difficult to re-warm.