Training Dynamics Impact Quantization Degradation

Albert Catalan-Tatjer[†‡*] **Niccolò Ajroldi**[†] **Jonas Geiping**[†‡]

[†]ELLIS Institute Tübingen
[‡]Max Planck Institute for Intelligent Systems & Tübingen AI Center

Abstract

Despite its widespread use, little is understood about what makes large language models more — or less — robust to quantization. To address this question, we study the degradation induced by post-training quantization (PTQ) in language modeling, analyzing open-source training trajectories of models up to 3 billion parameters and 11 trillion tokens. Furthermore, we validate our analysis by pretraining 160M-parameter models on up to 100B tokens. Our findings reveal that, post-training quantization robustness is driven by a complex interplay between learning rate decay and validation loss. In particular, as learning rate decays, validation loss and quantization error diverge, mostly independent of the amount of training data. As a consequence, we present two examples of interventions on the training dynamics that modulate quantization error, sometimes favorably. Namely, (1) for comparable validation loss, higher learning rates can lead to smaller quantization error; (2) weight averaging approximates learning rate decay favorably in some settings.

1 Introduction

The present of deep learning is already low-bit [NVIDIA, 2025]. Quantization has emerged as an horizontal technique that can be plugged in different parts of the deep learning pipeline - pretraining [Peng et al., 2023, Tseng et al., 2025, Wang et al., 2025], optimizer states [Dettmers et al., 2022b, Li et al., 2023, Huang et al., 2025], post-training quantization (PTQ) [Frantar et al., 2023, Lin et al., 2024, Tseng et al., 2024] - to unlock low-bit primitive throughput and memory gains. It is used in popular language models such as the deepseek models [DeepSeek-AI et al., 2025] or gpt-oss [OpenAI et al., 2025]. In general, quantization can be summarized as mapping full-precision (FP) values to low-precision representations while preserving accuracy as much as possible. Common strategies include scaling [Xiao et al., 2024], rotating [Frantar et al., 2023], grouping [Lin et al., 2024], or indexing in codebooks [Tseng et al., 2024]. Despite the widespread use of post-training quantization (PTQ), there is limited understanding of the principles that govern its sensitivity.

Two recent studies, one by Kumar et al. [2024] and the other by Ouyang et al. [2024], claim that post-training quantization becomes less effective as models are trained on more data, arguing that quantization error increases with the number of tokens. However, both studies depend on studies performed following the same training recipe, and training dynamics are not a part of the resulting scaling laws. We focus on this particular issue, in fact, in Section 2 we show two examples of training dynamics having a larger effect on quantization robustness than data budget.

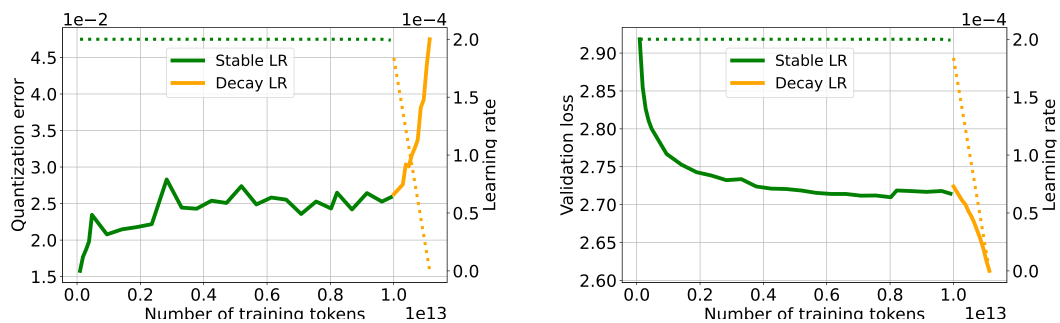
Our findings reveal that as learning rate decays, the validation loss decreases and coincidentally, quantization error surges, unveiling a deeper connection between PTQ robustness and training dynamics. We analyze this effect across multiple training runs and open-source models, exploring a range of hyperparameter settings and training horizons. Finally, we show that weight averaging [Izmailov

[*]albert.catalan-tatjer@tue.ellis.eu

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: NeurIPS 2025 Workshop on Efficient Reasoning.

et al., 2019, Kaddour, 2022] successfully modulates the full-precision to quantized validation loss trade-off. We summarize our contributions as follows:

1. We decouple the influence of training data from quantization degradation, showing that robustness is not primarily determined by scale alone.
2. We examine learning rate decay and validation loss, suggesting their interplay as critical factors underlying PTQ robustness.
3. We intervene on the learning rate magnitude and find that, in our settings, higher learning rates result in lower quantization error for the same validation loss.
4. We identify weight averaging as a substitute of learning rate decay with favorable post-training quantization robustness. In some cases, leading to a lower validation loss for the quantized weights.



(a) Quantization error vs training tokens.

(b) Validation loss vs training tokens.

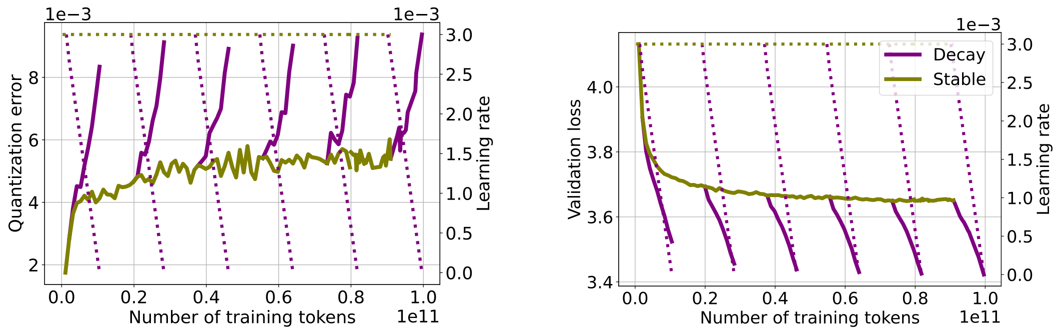
Figure 1: Evolution of quantization error and validation loss during training for SmolLM3 [Bakouch et al., 2025]. We show the evolution of quantization error during training in Figure 1a, where solid lines show the difference in validation loss between full precision checkpoints and their quantized counterparts during training, while dotted curves trace the learning rate evolution. Curves during the **stable** phase ($\eta = 2e^{-4}$) are reported in green, and during the **decay** phase in yellow. We quantize to 4 bits using GPTQ [Frantar et al., 2023]. In Figure 1b we show the evolution of validation loss during training. The learning rate schedule also appears as dotted lines. We observe that as the learning rate decay triggers a surge in the quantization error and a decline in the validation loss.

2 Training Dynamics and Quantization

SmolLM-3B. We track the evolution of quantization error along the training trajectory of SmolLM3 [Bakouch et al., 2025]. This 3B-parameter language model is well suited for our study due to its long training horizon of 11T tokens, and adoption of a Warmup–Stable–Decay (WSD) schedule [Hu et al., 2024, Haegele et al., 2024], which conveniently splits training into constant-LR stable phase and a linear-decay phase, thereby isolating the influence of learning rate dynamics on quantization error.

Figure 1a shows quantization error — measured as the difference in validation loss between each checkpoint and its quantized counterpart — alongside the learning rate schedule. We primarily use GPTQ [Frantar et al., 2023] and report consistent results across other backends, models, bit-widths, and methods in Appendix B. Whereas prior work argued that quantization error increases with the number of tokens [Kumar et al., 2024, Ouyang et al., 2024], we observe a different pattern: after an initial rise during the first 20B tokens, error grows only slowly throughout the stable phase despite the increasing number of tokens, but surges sharply in the decay phase as the learning rate decreases. Figure 1b demonstrates that the validation loss follows a similar - albeit inverse - curve than that of the quantization error. While the learning rate itself may not directly cause this degradation, this observation suggests a deeper connection between optimization dynamics and quantization performance.

Controlled experiments. To gain a deeper understanding and isolate unknown idiosyncrasies of online training trajectories, we pretrain several Pythia [Biderman et al., 2023] models of 160M parameter on FineWedEdu [Penedo et al., 2024], varying learning rate, learning rate schedule, and training horizons. We refer to Appendix A for more details on the training procedure. Figure 2 shows results across a range of token budgets, obtained by decaying the learning rate at different points during training. We track the evolution of quantization error across several decay phases, and observe that, despite training durations ranging from 10B to 100B tokens, models achieve comparable quantization error after learning rate decay, highlighting the independence between the number of ingested tokens and PTQ robustness.



(a) Quantization error vs training tokens.

(b) Validation loss vs training tokens.

Figure 2: Quantization error across different training durations for Pythia-160M on FineWebEdu. We use WSD schedule, training up to 100B tokens and performing additional cooldowns after 12B, 28B, 46B, 64B, 82B tokens. Figure 2a shows the evolution of quantization error during training for the same model with different token budgets, and Figure 2b reports the corresponding evolution of validation loss. We show the learning rate as dotted lines in both figures to highlight the decay phase for each training runs. Despite varying the training set scales, all models show comparable quantization error after cooldown, highlighting that error spikes are driven by learning rate rather than token budget.

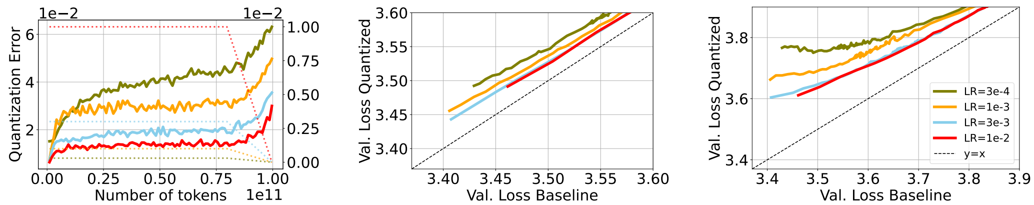
3 Interventions

Having shown the connection between training dynamics and quantization error, we present two examples where intervening in the optimization process can modulate PTQ robustness and, in some settings, even yield better quantized models.

Learning rate balances quantization error. In Figure 3, we ablate different choices of top learning rate to study their impact on quantization. Figure 3a shows that higher learning rates consistently lead to smaller errors, with curves inversely ordered by rate magnitude. Figure 3b and Figure 3c further report full-precision versus 4-bit and 3-bit quantized validation losses. These parametric curves capture quantization error relative to total validation loss: perfect quantization would lie on the $x = y$ bisector, with deviations measuring the error. Notably, comparing the curves with LR $1\mathrm{e}{-3}$ and $3\mathrm{e}{-3}$ shows that, at similar validation loss, the larger rate achieves better low-bit quantization at no apparent cost. The findings suggest that, for comparable validation loss, employing a larger learning rate is preferable, as it enhances low-bit quantization performance.

Weight Averaging can reduce quantization degradation. Given the detrimental effect of learning rate decay on quantization performance, a natural question is whether weight averaging could serve as an alternative and mitigate its negative impact. Intuitively, averaging parameters along the training trajectory reduces noise and can act as a proxy for learning rate decay. Prior work derived equivalent averaging schemes for common learning rate schedules under SGD [Sandler et al., 2023], and later studies showed that averaging can greatly improve performance over constant learning rate training [Haegele et al., 2024, Ajroldi et al., 2025], though still falling short of learning rate decay. Nevertheless, its effect on PTQ robustness remains unexplored, despite its simplicity, negligible cost, and compatibility with existing pipelines.

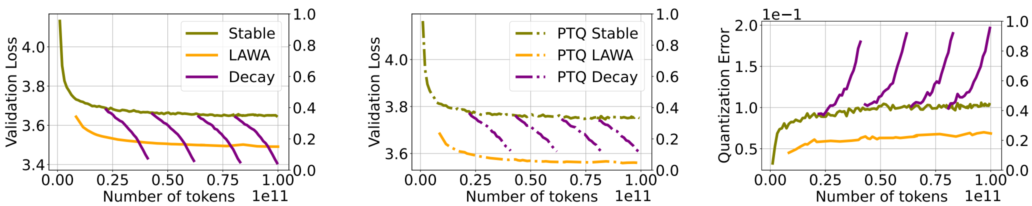
We pretrain a 160M-parameter Pythia model on 100B tokens with a constant learning rate and compare Latest Weight Averaging (LAWA) [Kaddour, 2022] against several intermediate learning-rate cooldowns. As observed in prior work [Ajroldi et al., 2025], in the full-precision setting (Figure 4a), LAWA yields better checkpoints than constant learning rate but does not reach the



(a) Quantization error. **(b)** FP to 4 bit PTQ validation loss. **(c)** FP to 3 bit PTQ validation loss.

Figure 3: Larger learning rates lead to lower quantization error. Figure 3a displays the quantization error achieved by fixing the training recipe and varying the learning rate. We observe that quantization error decreases when employing higher learning rates. Furthermore, Figure 3b and 3c show that, at similar validation loss, larger learning rates achieve better low-bit quantization at no apparent cost.

performance of intermediate cooldowns. In contrast, for quantized models (Figure 4b), checkpoints obtained through weight averaging match—or even surpass—the performance of those trained with learning-rate decay.



(a) FP Validation Loss. **(b)** 3 bit PTQ validation loss **(c)** Quantization Error.

Figure 4: Weight averaging and quantization error. We show the validation performance and quantization error along the training trajectory of a Pythia 160M parameter model trained for up to 100B tokens at constant learning rate, and compare intermediate learning rate cooldowns with weight averaging of checkpoints collected during the stable phase. We report the validation performance of the full-precision model (Figure 4a), the 3-bit quantized model (Figure 4b), and their difference (Figure 4c). Whereas LAWA falls short of learning-rate decay in the full-precision setting, its 3-bit PTQ performance yields lower validation loss than all cooldowns, demonstrating a successful setting for LAWA.

4 Conclusion

We conduct a systematic investigation of how training interventions affect quantization degradation in language models under controlled experimental configurations. First, we find that with all other hyperparameters fixed, learning rate magnitude alone determines quantization error. Therefore, in a scenario where two training runs attain comparable validation loss, we recommend breaking the tie choosing the one with higher learning rate, for its expected enhanced quantization performance. Secondly, we study replacing learning rate decay by weight averaging, using LAWA. Although recent work suggests that it does not bridge the validation loss gap, we find that LAWA is more robust to PTQ. In fact, for some lower bit settings, we observe that LAWA outperforms learning rate decay. These examples are concrete cases in which quantization error is noticeably changed through changes in training dynamics, leading us to argue that training dynamics should be carefully investigated for favorable quantization performance.

Nevertheless, the mechanisms through which learning rates and weight averaging affect quantization performance remain unclear. As a result, whether a predictive model of quantization degradation is within reach, or what additional factors may be at play, is still an open question.

Overall, we end with an optimistic note. Our findings indicate that quantization degradation stems from an intricate relationship between training dynamics and learning rate decay. As a result, we find that rather than being an unavoidable consequence of training data scale, it can be acted upon with existing tools and mechanisms, which are especially beneficial for low-bit quantization.

References

Niccolò Ajroldi. plainlm: Language model pretraining in pytorch. https://github.com/Niccolo-Ajroldi/plainLM, 2024.

Niccolò Ajroldi, Antonio Orvieto, and Jonas Geiping. When, where and why to average weights? In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=JN8O01IZYR.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clementine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner, 2025. URL https://huggingface.co/blog/smollm3.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling, May 2023. URL http://arxiv.org/abs/2304.01373. arXiv:2304.01373 [cs].

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. DeepSeek-V3 Technical Report, February 2025. URL http://arxiv.org/abs/2412.19437. arXiv:2412.19437 [cs].

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale, November 2022a. URL http://arxiv.org/abs/2208.07339. arXiv:2208.07339 [cs].

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 8-bit Optimizers via Block-wise Quantization, June 2022b. URL http://arxiv.org/abs/2110.02861. arXiv:2110.02861 [cs].

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers, March 2023. URL http://arxiv.org/abs/2210.17323. arXiv:2210.17323 [cs].

Alexander Haegele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations, 2024. URL https://arxiv.org/abs/2405.18392.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm: Unveiling the potential of small language models with scalable training strategies, 2024. URL https://arxiv.org/abs/2404.06395.

Tianjin Huang, Haotian Hu, Zhenyu Zhang, Gaojie Jin, Xiang Li, Li Shen, Tianlong Chen, Lu Liu, Qingsong Wen, Zhangyang Wang, and Shiwei Liu. Stable-SPAM: How to Train in 4-Bit More Stably than 16-Bit Adam, April 2025. URL http://arxiv.org/abs/2502.17055. arXiv:2502.17055 [cs].

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization, 2019.

Jean Kaddour. Stop Wasting My Time! Saving Days of ImageNet and BERT Training with Latest Weight Averaging, October 2022. arXiv:2209.14981 [cs, stat].

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL https://api.semanticscholar.org/CorpusID:6628106.

Tanishq Kumar, Zachary Ankner, Benjamin F. Spector, Blake Bordelon, Niklas Muennighoff, Mansheej Paul, Cengiz Pehlevan, Christopher Ré, and Aditi Raghunathan. Scaling Laws for Precision, November 2024. URL http://arxiv.org/abs/2411.04330. arXiv:2411.04330.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Bingrui Li, Jianfei Chen, and Jun Zhu. Memory efficient optimizers with 4-bit states, 2023. URL https://arxiv.org/abs/2309.01507.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. Gptqv2: Efficient finetuning-free quantization for asymmetric calibration. *arXiv preprint arXiv:2504.02692*, 2025.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration, July 2024. URL http://arxiv.org/abs/2306.00978. arXiv:2306.00978.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

ModelCloud.ai and qubitium@modelcloud.ai. Gptqmodel. https://github.com/modelcloud/gptqmodel, 2024.

NVIDIA. Introducing NVFP4 for efficient and accurate low-precision inference. https://developer.nvidia.com/blog/introducing-nvfp4-for-efficient-and-accurate-low-precision-inference/, June 2025. NVIDIA Technical Blog.

OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park

Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b Model Card, August 2025. URL http://arxiv.org/abs/2508.10925. arXiv:2508.10925 [cs].

Xu Ouyang, Tao Ge, Thomas Hartvigsen, Zhisong Zhang, Haitao Mi, and Dong Yu. Low-Bit Quantization Favors Undertrained LLMs: Scaling Laws for Quantized LLMs with 100T Training Tokens, November 2024. URL http://arxiv.org/abs/2411.17691. arXiv:2411.17691 [cs].

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Houwen Peng, Kan Wu, Yixuan Wei, Guoshuai Zhao, Yuxiang Yang, Ze Liu, Yifan Xiong, Ziyue Yang, Bolin Ni, Jingcheng Hu, Ruihang Li, Miaosen Zhang, Chen Li, Jia Ning, Ruizhe Wang, Zheng Zhang, Shuguang Liu, Joe Chau, Han Hu, and Peng Cheng. FP8-LM: Training FP8 Large Language Models, December 2023. URL http://arxiv.org/abs/2310.18313. arXiv:2310.18313 [cs].

Mark Sandler, Andrey Zhmoginov, Max Vladymyrov, and Nolan Miller. Training trajectories, mini-batch losses and the curious role of the learning rate, February 2023. arXiv:2301.02312 [cs].

Albert Tseng, Qingyao Sun, David Hou, and Christopher De Sa. QTIP: Quantization with Trellises and Incoherence Processing, June 2024. URL http://arxiv.org/abs/2406.11235. arXiv:2406.11235 [cs].

Albert Tseng, Tao Yu, and Youngsuk Park. Training LLMs with MXFP4, August 2025. URL http://arxiv.org/abs/2502.20586. arXiv:2502.20586 [cs].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Ruizhe Wang, Yeyun Gong, Xiao Liu, Guoshuai Zhao, Ziyue Yang, Baining Guo, Zhengjun Zha, and Peng Cheng. Optimizing Large Language Model Training Using FP4 Quantization, May 2025. URL http://arxiv.org/abs/2501.17116. arXiv:2501.17116 [cs].

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 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, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models, 2024. URL https://arxiv.org/abs/2211.10438.

A Replicability

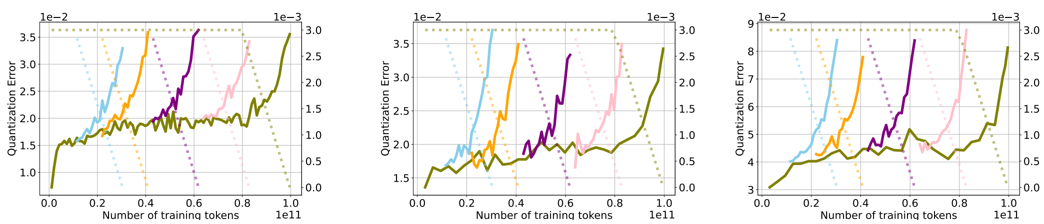
Quantization We quantize all the checkpoints with from huggingface transformers Wolf et al. [2020] with different quantization backends. For GPTQ [Frantar et al., 2023, Li et al., 2025] we use GPTQModel ModelCloud.ai and qubitium@modelcloud.ai [2024]. For AWQ Lin et al. [2024] we use Kwon et al. [2023]. And for LLM.int8() Dettmers et al. [2022a] we use HuggingFace Wolf et al. [2020].

Pretrain We use the open source codebase from Ajroldi [2024] to pretrain Pythia-160M parameter transformer Biderman et al. [2023], Vaswani et al. [2023] on language modeling, training up to 100 billion tokens of FineWebEdu Penedo et al. [2024] on up to 8 A100 80GB GPUs. We employ a sequence length of 2048 and batch size of 0.5M tokens. We use cross-entropy loss and employ Adam [Kingma and Ba, 2014] with decoupled weight decay [Loshchilov and Hutter, 2019] of 0.1 and gradient clipping of 1.

Evaluation We evaluate on a held-out set of refinedweb with...

B Quantization backbones

Our results are centered around GPTQ Frantar et al. [2023] a popular and easy-to-use quantization method that works off-the-shelf for new models with minimal engineering overhead. However, we replicate figure 2 with LLM.int8() Dettmers et al. [2022a] and AWQ Lin et al. [2024] to investigate whether our observed phenomena are particular to GPTQ or to PTQ as a whole. We show this results in figure, where relationship between the factors under study appears to be consistent.

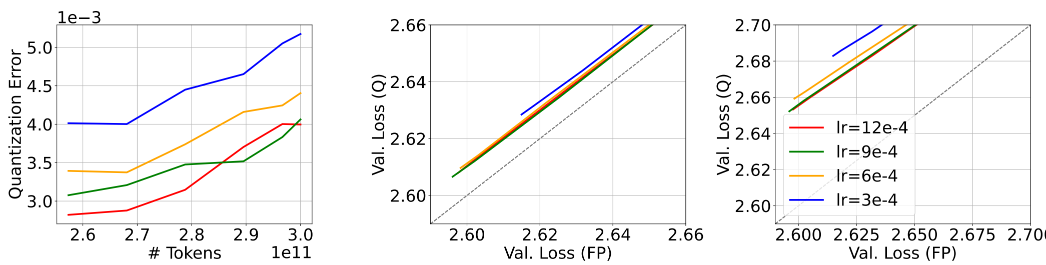


(a) GPTQ Frantar et al. [2023]. **(b)** AWQ Lin et al. [2024]. **(c)** LLM.int8Dettmers et al. [2022a]

Figure 5: Quantization error on different 4 bit quantization backends.

C Large scale experiments

We replicate the study of Figure 3 with OLMo2 7 billion parameter model trained for 300 billion tokens with cosine decay, with linear annealing during 50 billion tokens additionally. The learning rates of the sweep are $\{12e^{-4}, 9e^{-4}, 6e^{-4}, 3e^{-4}\}$. We observe the same results. The red, green and orange lines achieve comparable performance, and the full-precision loss to quantized loss curves are sorted by learning rate magnitude, from lowest to highest, indicating that larger learning rates learn a model that is more robust to post-training quantization.



(a) Quantization error. **(b)** FP to 4-bit PTQ val. loss. **(c)** FP to 4-bit PTQ val. loss

Figure 6: Quantization error on different 4 bit quantization backends.