Loss and Optimizer as Two Essential Mechanisms Behind Knowledge Distillation

Satoki Ishikawa ISHIKAWA@RIO.SCRC.IIR.ISCT.AC.JP
Institute of Science Tokyo

Sameer Satish Deshmukh SAMEER.DESHMUKH@RIO.GSIC.TITECH.AC.JP
Fujitsu Limited

Sakina Fatima SAKINA.FATIMA@FUJITSU.COM
Fujitsu Limited

Takumi Honda HONDA.TAKUMI@FUJITSU.COM
Fujitsu Limited

Rio Yokota RIOYOKOTA@RIO.SCRC.IIR.ISCT.AC.JP
Institute of Science Tokyo

Abstract

Knowledge distillation (KD) is typically improved through better divergence functions between teacher and student models, while the role of the optimizer has received less attention. In this paper, we argue that optimizer choice can also play an important role in KD, since both the divergence and the optimizer reshape the training dynamics under a finite training budget. On CIFAR-100 distillation, replacing SGD with SOAP often improves accuracy more than replacing KD with ABKD. On GPT-2 distillation with Dolly, preconditioned gradients provide little improvement, whereas switching from KD to GKD or ABKD increases the critical batch size and enables large-batch training, similar to the effect of preconditioned GD in large-batch training. We hope that this paper serves as a first step toward understanding the role of preconditioned GD in distillation.

1. Introduction

Knowledge distillation (KD)[16] enhances the performance of small student models by transferring knowledge from a resource-intensive teacher model. The student model's probability distribution p is trained to follow the probability distribution q_θ of the teacher model. The loss function plays a crucial role in how well the student model can match the probability distribution of the teacher model. Divergence functions such as variants of Kullback-Leibler divergence have become increasingly common as a choice of loss function[1, 11, 14, 21, 23, 28].

In principle, a loss function using Kullback-Leibler divergence should lead to convergence of the student and teacher probability distributions provided that the student model has sufficient expressive capacity to represent the teacher's distribution, and the student model is trained for a sufficiently long time [32, 37]. However, even when the two objectives converge to the same distribution in function space, they may arrive at different solutions in weight space, leading to different generalization error. Moreover, in practice, where the available training budget is limited, differences in convergence speed arising from distinct training dynamics also become a critical factor that de-

termine the final performance. From this perspective, the choice of a divergence in KD can be attributed not only to the divergence itself but also to the training dynamics it induces.

A closely analogous phenomenon arises in preconditioned GD methods such as Shampoo [15] or Muon [18], where it is the optimization trajectory, rather than the optimization objective alone, that shapes the final model parameters. Even when we train by preconditioned GD, the explicit loss function remains unchanged, but its training trajectory differs, and as a result, the model converges to solutions with different generalization performance [2, 19]. In addition, because preconditioned methods use richer information via the second order curvature in a single update step, they can improve the convergence of the training loss [5, 24, 29]. In short, the optimizer complements the divergence function in KD: both reshape the training dynamics and thereby the final solution. In spite of this, the role of the optimizer in KD has received less attention than role of the divergence.

Motivated by this analogy, we empirically investigate the interplay of the choice of optimizer and divergence in distillation. Our findings are as follows.

1. On vision tasks, we identify cases in which the performance gain obtained by changing the divergence of loss function is nearly identical to the gain obtained by changing the optimizer.

2. On the dolly language task [12], neither SOAP ;[35] nor Muon[18] yielded significant gains. Although both are known to enable large-batch training here, they did not. Instead, switching from forward KL to GKD or ABKD was what made larger batch training possible.

Together, these results show that divergence design and optimizer choice can play analogous roles in shaping optimization trajectories under finite training budgets.

1.1. Related Works

Knowledge Distillation in LLM Knowledge distillation (KD) was originally proposed by Hinton et al. [16] as a framework for transferring the soft output distribution of a large teacher model into a smaller student model, and has since become a fundamental approach for model compression [38]. Divergence between two probability distributions can be achieved primarily through mode-seeking [1, 14, 21, 23] or mean-seeking [11, 16, 28] divergence functions. While their solutions coincide under certain conditions, their training dynamics differ substantially [32, 37]. Furthermore, several works generalize the Kullback-Leibler divergence; for instance, ABKD uses the α-β-divergence as such a generalization [36].

Preconditioned Gradient Descent Preconditioned gradient descent (GD) accelerates optimization by modifying gradient updates with second-order statistics accumulated during training. A representative example is Shampoo [15], which approximates full-matrix AdaGrad with a Kronecker-factored preconditioner and has been shown to accelerate training [3, 7, 20]. SOAP [35] further improves Shampoo by reducing computational overhead and stabilizing behavior through Adam-style updates in the preconditioner's eigenbasis. A related class is Spectral GD [8], which accelerates training by normalizing or orthogonalizing gradient matrices. Muon [18] extends this by applying momentum to orthogonalized updates and has recently attracted attention for large language model training. Although Shampoo and Spectral GD arise from different motivations, their updates are closely related [6, 30].

The benefits of preconditioned GD are not limited to faster per-step optimization. In large-batch training, increasing the batch size improves parallel efficiency only up to the critical batch

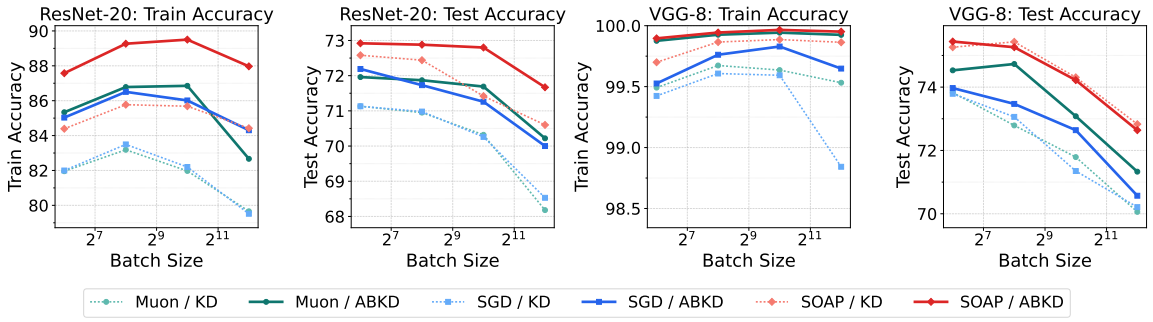

Figure 1: **SOAP achieves high accuracy in distillation for vision tasks.** We conducted distillation experiments on CIFAR-100 using ResNet and VGG. Except for ResNet with a batch size of 1024, switching the optimizer from SGD to SOAP while retaining KD yielded a larger accuracy gain than replacing KD with ABKD while retaining SGD in all settings. For ABKD, we set $\alpha = 0.5$ and $\beta = 0.5$, which corresponds to the Hellinger distance.

size, beyond which additional samples yield diminishing returns [26, 27, 40]. By improving the update direction, preconditioned GD can increase this critical batch size, enabling more efficient large-batch training and thereby reducing training time at the system level [17, 29, 31, 39].

2. Experiments

2.1. SOAP Boosts Distillation Performance in Vision Tasks

We conducted distillation experiments on CIFAR-100 [22]. Specifically, we evaluated distillation from ResNet56 to ResNet20 and from VGG13[33] to VGG8[34], with two distillation methods: standard KD (forward Kullback-Leibler divergence) and ABKD. As shown in Figure 1, SOAP achieves higher accuracy than distillation with SGD. Strikingly, in most cases, using KD with SOAP yields higher accuracy than using ABKD with SGD. Muon outperforms SGD but falls short of SOAP. These results suggest that in the vision distillation setting, the performance gains from improving the divergence are primarily driven by enhanced trainability.

We further observe that when training ResNet with ABKD + SOAP, accuracy remains essentially unchanged up to a batch size of 1024, whereas with KD + SOAP, accuracy at batch size 1024 degrades compared to batch size 256. This indicates that an additional benefit of using ABKD over KD is the ability to train with larger batch sizes.

In Figure 2, we vary the values of α and β in ABKD. We observe that SOAP achieves the highest accuracy regardless of the values of α and β. We also observe that the optimal values of α and β differ slightly across optimizers. As a side note, there is a regime where larger β leads to higher train accuracy but lower test accuracy. In the low-β regime, the student can better exploit the soft-label information from the teacher distribution; this acts as a form of regularization, so that even as train accuracy decreases, test accuracy improves.

2.2. Preconditioned GD Shows Little to No Effect in Language Tasks with Dolly Dataset

Figure 5 reports our GPT distillation experiments, showing the average accuracy across five datasets for ABKD, KD, GKD, and SFT. The results are obtained by first training the student and teacher

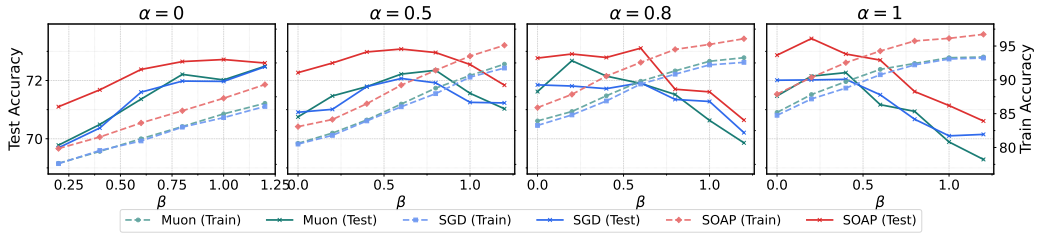

Figure 2: **Test Accuracy (solid line) and Train Accuracy (dashed line) in ABKD with different α and β.** We conducted distillation experiments from ResNet56 to ResNet20 on CIFAR-100. Regardless of the values of alpha and beta, SOAP consistently achieves the highest train and test accuracy. Note that in this task, larger values of beta yield higher train accuracy, but test accuracy is lower in this regime.

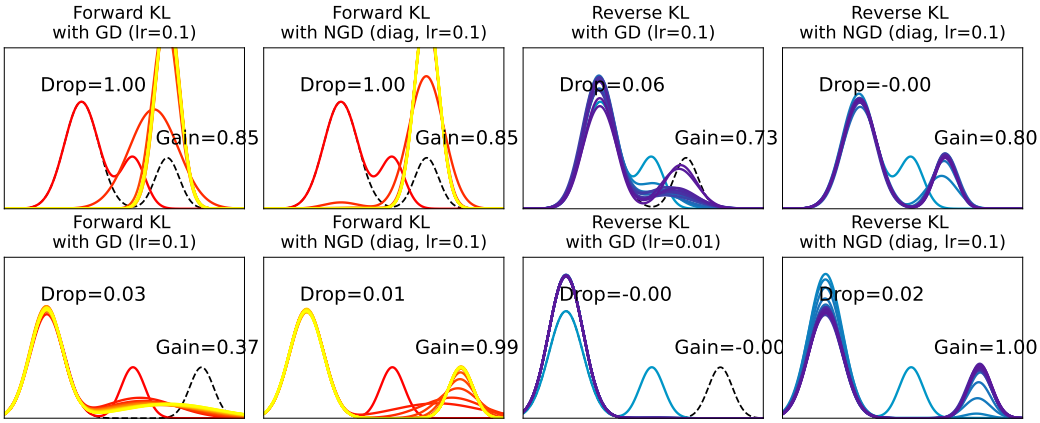

Figure 3: (top) Teacher samples are drawn only from the right Gaussian component; consequently, Forward-KL training is biased toward the right mode and tends to retain the left mode. This is similar to a continual learning setup in which modes are learned sequentially [9]. (bottom) Teacher samples are drawn from the full two-component mixture (both left and right modes), so training fits the student while observing both modes.

models with the initialization described in Section B.1, and each run is trained for 20 epochs. In this GPT distillation setting, SOAP and Muon yield little to no improvement in accuracy; instead, changing the divergence from KD (forward KL) to GKD or ABKD proves far more effective. Furthermore, while KD's accuracy decreases monotonically as the batch size grows, GKD maintains nearly constant accuracy up to a batch size of $32{,}768$. This suggests that GKD has a larger critical batch size than KD, enabling training at substantially larger batch sizes.

2.3. Toy Example with Mixture of Gaussians

In the CIFAR-100 vision experiments, changing the optimizer had a large effect, whereas in the language experiments it does not change so much. We show that these two distinct regimes also emerge in a toy model with Gaussian mixtures, a common setting for analyzing distillation [10]. Both teacher and student are two-component GMMs of the form $q(x) = \pi\mathcal{N}(x \mid \mu_1, \sigma_1) + (1 -$

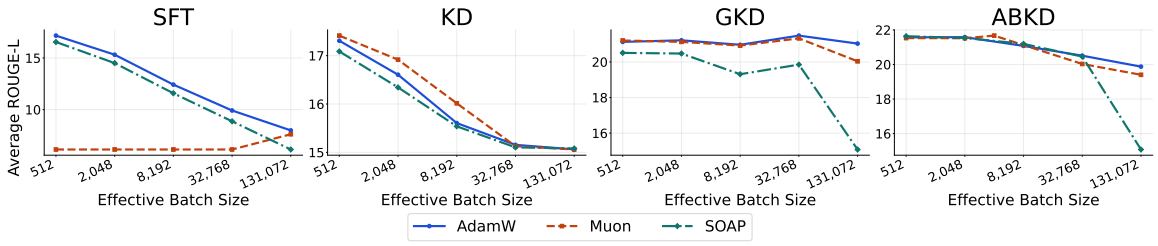

Figure 5: **ABKD and GKD perform well across different batch size.** This graph shows how accuracy changes as the batch size is varied in the distillation with GPT. Increasing the batch size from 512 to 131,072 leads to an accuracy drop of nearly 10% for SFT and around 2% for KD. GKD with AdamW is the most robust to batch size scaling, exhibiting almost no change in accuracy.

$\pi)\,\mathcal{N}(x \mid \mu_2, \sigma_2)$, and we compare standard GD against diagonal NGD under two divergences (Forward / Reverse KL) and two sampling regimes (teacher samples drawn only from the right component, or from the full mixture). Figure 3 shows that the gap between GD and diagonal NGD is highly setting-dependent. Under Forward KL, or under Reverse KL with single-mode sampling, GD and diagonal NGD behave qualitatively similarly: both either retain the existing left mode (Reverse KL, since updates are confined to regions covered by teacher samples) or forget it (Forward KL, which biases the student toward the observed right mode). In contrast, under Reverse KL with full-mixture sampling, GD becomes trapped in a single-mode solution and fails to acquire the new mode, whereas NGD recovers both modes. In this case, using diag-NGD becomes critically important like in vision experiments.

The vector-field analysis of the simplified case where only π and μ_2 are trained (Figure 4) clarifies the mechanism of Reverse KL: GD first drives π toward 1 while barely moving μ_2, converging to a single-mode fixed point, while NGD instead prioritizes reducing the error in μ_2 and steers the trajectory toward a two-mode solution. Together, these observations show that preconditioning can qualitatively alter the solution selected under a finite training budget, but the magnitude of this effect depends on the task settings.

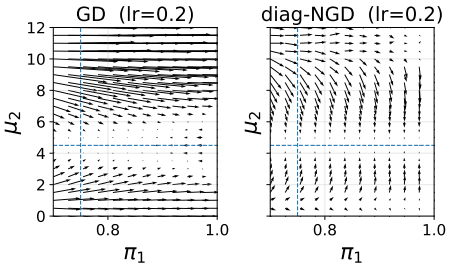

Figure 4: **Vector Fields for Reverse KL Training.**

3. Discussions

Exploration of New Optimizers. In the training of vision models, K-FAC has long been a popular preconditioned GD for accelerating training [13, 25]. For language models such as GPT, however, K-FAC has proven less effective [4], and Muon and SOAP have instead emerged as compelling alternatives, attracting considerable attention in recent years. We expect a similar trajectory for distillation: just as SOAP proved highly effective for vision-model distillation in our experiments, developing new optimizers for GPT-style distillation is likely to be a promising direction for accelerating training.

Numerical Stability of Optimizers. Any such optimizer, however, must also be numerically robust. Figure 6 shows the results when training is performed in FP16 instead of BF16. Performance degrades across the board, but the drop is particularly severe for Muon: with ABKD or GKD, accuracy collapses entirely at BS = 2048. Moreover, SOAP achieves the highest accuracy on ABKD and this differs from the BF16 setting. These findings indicate that whether an optimizer improves accuracy depends critically on its robustness to numerical precision, and that this sensitivity grows with batch size. Consequently, future GPT distillation methods targeting large-batch training will need to account for stability under reduced numerical precision as well.

References

[1] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

[2] Shun-ichi Amari, Jimmy Ba, Roger Grosse, Xuechen Li, Atsushi Nitanda, Taiji Suzuki, Denny Wu, and Ji Xu. When does preconditioning help or hurt generalization? *arXiv preprint arXiv:2006.10732*, 2020.

[3] Rohan Anil, Vineet Gupta, Tomer Koren, Kevin Regan, and Yoram Singer. Scalable second order optimization for deep learning. *arXiv preprint arXiv:2002.09018*, 2020.

[4] Juhan Bae, Paul Vicol, Jeff Z HaoChen, and Roger B Grosse. Amortized proximal optimization. *Advances in Neural Information Processing Systems*, 35:8982–8997, 2022.

[5] Alberto Bernacchia, Máté Lengyel, and Guillaume Hennequin. Exact natural gradient in deep linear networks and its application to the nonlinear case. *Advances in Neural Information Processing Systems*, 31, 2018.

[6] Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024.

[7] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10925–10934, 2022.

[8] David E Carlson, Edo Collins, Ya-Ping Hsieh, Lawrence Carin, and Volkan Cevher. Preconditioned spectral descent for deep learning. *Advances in neural information processing systems*, 28, 2015.

[9] Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. *arXiv preprint arXiv:2510.18874*, 2025.

[10] Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting, 2025.

[11] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802, 2019.

[12] M. Conover, M. Hayes, A. Mathur, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, and R. Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm. Company Blog of Databricks, 2023.

[13] Roger Grosse and James Martens. A kronecker-factored approximate fisher matrix for convolution layers. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2016.

[14] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The twelfth international conference on learning representations*, 2024.

[15] Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pages 1842–1850. PMLR, 2018.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[17] Satoki Ishikawa and Rio Yokota. When does second-order optimization speed up training? In *The Second Tiny Papers Track at ICLR 2024*, 2024.

[18] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

[19] Ryo Karakida and Kazuki Osawa. Understanding approximate fisher information for fast convergence of natural gradient descent in wide neural networks. *Advances in neural information processing systems*, 33:10891–10901, 2020.

[20] Priya Kasimbeg, Frank Schneider, Runa Eschenhagen, Juhan Bae, Chandramouli Shama Sastry, Mark Saroufim, BOYUAN FENG, Less Wright, Edward Z. Yang, Zachary Nado, Sourabh Medapati, Philipp Hennig, Michael Rabbat, and George E. Dahl. Accelerating neural network training: An analysis of the algoperf competition. In *The Thirteenth International Conference on Learning Representations*, 2025.

[21] Gyeongman Kim, Doohyuk Jang, and Eunho Yang. Promptkd: Distilling student-friendly knowledge for generative language models via prompt tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6266–6282, 2024.

[22] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, April 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. Technical Report.

[23] Hyoje Lee, Yeachan Park, Hyun Seo, and Myungjoo Kang. Self-knowledge distillation via dropout. *Computer Vision and Image Understanding*, 233:103720, 2023.

[24] Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.

[25] James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417. PMLR, 2015.

[26] Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

[27] William Merrill, Shane Arora, Dirk Groeneveld, and Hannaneh Hajishirzi. Critical batch size revisited: A simple empirical approach to large-batch language model training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=XUKUx7Xu89.

[28] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198, 2020.

[29] Kazuki Osawa, Yohei Tsuji, Yuichiro Ueno, Akira Naruse, Rio Yokota, and Satoshi Matsuoka. Large-scale distributed second-order optimization using kronecker-factored approximate curvature for deep convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12359–12367, 2019.

[30] Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained LMOs. In *Forty-second International Conference on Machine Learning*, 2025.

[31] Naoki Sato, Hiroki Naganuma, and Hideaki Iiduka. Convergence bound and critical batch size of muon optimizer. *arXiv preprint arXiv:2507.01598*, 2025.

[32] Donghyeok Shin, Yeongmin Kim, Suhyeon Jo, Byeonghu Na, and Il chul Moon. AMid: Knowledge distillation for LLMs with α-mixture assistant distribution. In *The Fourteenth International Conference on Learning Representations*, 2026.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[34] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020.

[35] Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M. Kakade. SOAP: Improving and stabilizing shampoo using adam for language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025.

[36] Guanghui Wang, Zhiyong Yang, Zitai Wang, Shi Wang, Qianqian Xu, and Qingming Huang. ABKD: Pursuing a proper allocation of the probability mass in knowledge distillation via α-β-divergence. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=vt65VjJakt.

[37] Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking kullback-leibler divergence in knowledge distillation for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5737–5755, 2025.

[38] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[39] Guodong Zhang, Lala Li, Zachary Nado, James Martens, Sushant Sachdeva, George Dahl, Chris Shallue, and Roger B Grosse. Which algorithmic choices matter at which batch sizes? insights from a noisy quadratic model. *Advances in neural information processing systems*, 2019.

[40] Hanlin Zhang, Depen Morwani, Nikhil Vyas, Jingfeng Wu, Difan Zou, Udaya Ghai, Dean Foster, and Sham M Kakade. How does critical batch size scale in pre-training? In *The Thirteenth International Conference on Learning Representations*, 2024.

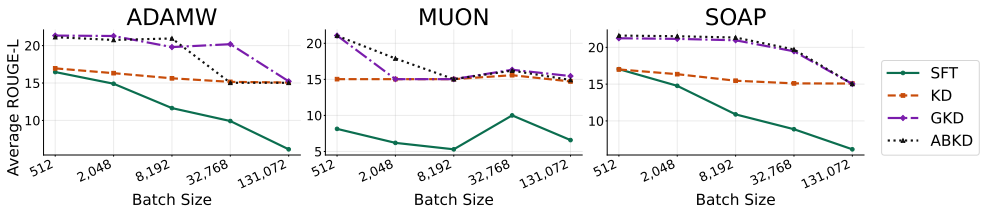

Figure 6: **Under FP16, Muon performs poorly in large-batch training.** While Figure 5 used BF16 for training, this figure shows results under FP16. Under FP16, Muon's accuracy drops sharply if we increase the batch size.

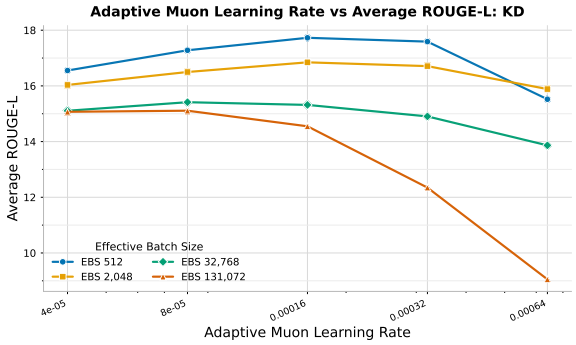

Figure 7: Learning rate vs. accuracy for adaptive muon using bfloat16 for KD. Adaptive muon does not show improvements over Muon with large batch sizes.

Appendix A. Additional Experimental Results

A.1. FP16 Training with Muon

Figure 6 reports GPT distillation on Dolly under FP16. In this setting, Muon becomes unstable as the batch size grows. In contrast, SOAP does not suffer such a severe accuracy drop. Since FP16 is rarely used for training in practice, Muon's weakness under FP16 is not in itself a serious concern. However, the fact that Muon becomes unstable when the exponent range is narrow will likely matter going forward, as it bears on the design of optimizers that remain robust under low precision.

A.2. Accuracy Evaluation of Adaptive Muon

A recent variant, AdaMuon, augments Muon with Adam-style element-wise adaptivity and has attracted growing attention. Figure 7 shows the results of training KD with AdaMuon. As the figure indicates, replacing Muon with AdaMuon does not noticeably increase the critical batch size.

A.3. Hessian Spectrum in the Distillation of GPT

Figure 8 visualizes how the Hessian spectrum varies across methods. SFT exhibits a much larger top eigenvalue, while the KD-based methods all show substantially smaller top eigenvalues. Among the KD-based methods themselves, however, the spectra are nearly indistinguishable.

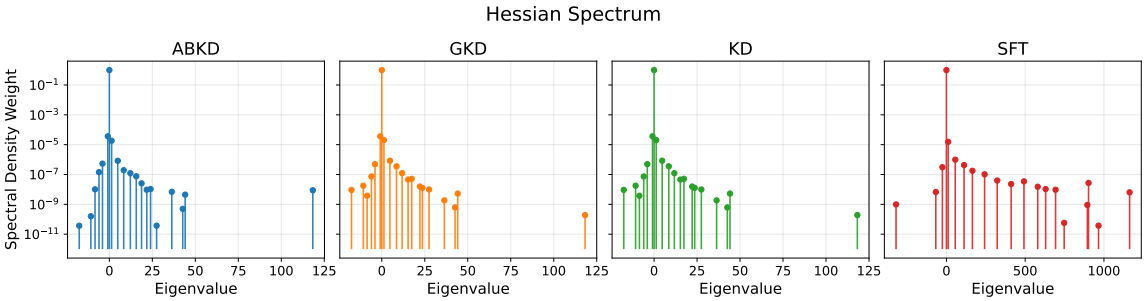

Figure 8: **Hessian spectra for GPT distillation.** SFT exhibits substantially larger eigenvalues than the distillation methods, while others produce comparable spectra.

Appendix B. Experimental Settings

B.1. Initialization of teacher and student

Our setup consists of a GPT2-XL (1.5B parameters) teacher model which is instruction tuned using the dolly dataset. This is done using a supervised fine tuning for 10 epochs, with a hyper parameter sweep where we vary the batch size and the learning rate, and taking the model checkpoint that shows the best validation error. The student model is a GPT2-small (0.1B parameters) student model which is instruction tuned using 3 epochs of dolly. Similar to the teacher model, we vary the learning rate and batch size and choose the checkpoint with the best validation loss. The dolly dataset, consisting of 15,000 rows, is split into a validation dataset of 1,000 rows and a training dataset of 14,000 rows. We use a maximum context length of 512 for both teacher and student models. A cosine annealing scheduler is used for varying the learning rate with 0 warmup iterations.

The AdamW optimizer is used for fine tuning both the student and the teacher. It uses a weight decay of 10^{-2}, gradient clipping set to 1.0, $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-8}$.

B.2. Figure 6

We perform the following in order to choose the best measurement:

- For AdamW, we vary the learing rate 2.5×10^{-6}, 5×10^{-6}, 10^{-5}, 2×10^{-5}, 4×10^{-5}, 8×10^{-5}, 16×10^{-5} and 32×10^{-5}. It uses a weight decay of 10^{-2}, gradient clipping set to 1.0, $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-8}$ for all measurements.
- For SOAP, we vary the learning rate 2.5×10^{-6}, 5×10^{-6}, 1×10^{-5}, 2×10^{-5}, 4×10^{-5}, 8×10^{-5} and 16×10^{-5}. For all measurements, we use a precondition frequency of 10, $\beta_1 = 0.95$, $\beta_2 = 0.95$, weight decay of 10^{-2}, maximum preconditioner dimension of $10,000$, $\epsilon = 10^{-8}$. All trainable parameters in all runs use SOAP.
- Muon is used on the 2D trainable parameters and AdamW is used on all other parameters. The Muon learning rate is varied as 10^{-5}, 10^{-4}, 10^{-3}, 2×10^{-5}, 2×10^{-4}, 2×10^{-3}, 4×10^{-4}, 4×10^{-3}, 8×10^{-4}, 8×10^{-3}, 1.6×10^{-3}, 1.6×10^{-2}, 3.2×10^{-3}, 3.2×10^{-2}, 6.4×10^{-3} and 6.4×10^{-2}. The AdamW learning rate for each corresponing Muon learning rate is a multiple of 1, 10 and 100 of the Muon learning rate. The Newton-Schulz co-efficients are

Batch size	Dolly Eval			Self-Instruct			Vicuna Eval			Super-Natural			Unnatural			Average		
	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon
2	26.50	26.67	25.74	13.05	13.91	13.05	16.22	15.73	14.55	27.97	28.07	25.80	25.96	26.64	24.47	21.94	22.20	20.72
4	26.97	26.67	26.69	13.47	13.73	13.35	16.66	16.25	16.19	27.62	27.99	27.15	26.01	26.56	25.75	22.15	22.24	21.82
8	26.72	26.84	26.55	13.13	13.29	13.07	16.80	16.68	17.02	27.58	27.91	27.02	25.75	26.42	25.76	22.00	22.23	21.88
16	26.65	26.69	26.57	12.90	13.08	12.98	15.87	16.40	16.70	27.57	27.59	26.55	25.94	25.85	25.99	21.79	21.92	21.76
32	27.17	26.62	26.29	12.69	12.68	12.36	16.56	16.98	15.49	27.45	27.68	26.24	25.55	25.36	24.96	21.88	21.86	21.07
64	26.48	26.49	26.65	12.60	12.51	12.84	16.20	16.82	16.90	27.44	27.59	26.57	25.69	25.75	25.62	21.68	21.83	21.72
128	26.49	26.51	26.36	12.62	12.73	11.74	16.16	16.57	16.02	27.19	27.43	26.01	25.65	25.11	23.89	21.62	21.67	20.80
256	26.35	26.31	26.43	12.87	12.61	12.66	16.44	16.72	16.95	27.58	27.70	26.91	25.55	25.45	26.27	21.76	21.76	21.85
512	26.42	26.53	26.07	12.46	12.54	12.24	16.01	16.69	16.04	27.32	27.54	25.81	25.54	25.55	25.26	21.55	21.77	21.08
1024	26.31	26.35	25.60	12.74	12.38	12.21	16.03	16.61	16.05	27.35	27.57	25.30	25.60	25.47	25.64	21.61	21.68	20.96
2048	25.93	25.88	21.14	12.70	12.70	10.58	16.74	16.58	17.03	27.22	27.19	21.30	25.83	25.41	20.42	21.68	21.55	18.10
4096	25.68	25.77	12.11	12.47	12.46	8.52	16.52	16.10	16.60	26.47	26.80	8.12	25.52	26.46	10.38	21.33	21.52	11.15
8192	25.17	25.25	7.93	12.33	12.37	5.60	16.15	16.52	15.16	26.64	26.89	4.44	25.91	26.07	5.60	21.24	21.42	7.75

Table 1: Evaluation results for ABKD with varying effective batch size.

Batch size	Dolly Eval			Self-Instruct			Vicuna Eval			Unnatural			Average		
	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon
2	24.22	22.39	21.43	10.50	9.71	9.60	14.82	12.90	13.41	20.11	19.38	16.20	17.41	16.10	15.16
4	24.49	22.75	22.92	10.66	9.68	10.41	15.07	13.49	13.58	18.91	18.54	18.48	17.28	16.12	16.35
8	24.30	22.53	23.75	10.41	9.94	11.11	15.08	13.63	14.73	18.34	18.31	20.70	17.03	16.10	17.57
16	24.09	23.66	24.15	10.32	10.24	10.96	15.11	13.88	14.94	18.94	19.22	20.00	17.11	16.75	17.51
32	24.07	23.91	24.11	10.06	10.26	10.80	14.85	14.38	14.85	18.64	18.62	19.65	16.90	16.79	17.35
64	23.43	23.86	24.30	10.46	10.19	10.70	14.45	15.03	15.32	18.72	18.13	20.13	16.76	16.80	17.61
128	23.70	23.64	22.59	10.34	10.37	10.23	14.71	14.46	14.60	17.82	18.15	19.35	16.64	16.66	16.69
256	24.15	22.95	24.29	10.45	10.42	10.53	15.17	14.57	15.61	18.60	18.10	18.89	17.09	16.51	17.33
512	22.54	22.47	17.65	10.14	10.35	7.21	14.49	14.45	11.12	17.76	17.81	12.83	16.23	16.27	12.20
1024	23.29	22.06	20.27	10.33	10.25	8.37	15.22	14.37	13.58	17.21	17.70	15.89	16.51	16.09	14.53
2048	21.85	21.55	10.35	9.66	9.84	5.33	14.09	14.31	10.75	17.14	17.12	5.93	15.69	15.70	8.09
4096	21.01	21.25	4.67	9.64	9.72	3.05	13.70	13.89	6.00	16.40	16.25	4.22	15.19	15.28	4.48
8192	20.81	20.77	3.43	9.21	9.22	2.46	13.66	13.89	5.43	16.67	16.29	3.38	15.09	15.04	3.68

Table 2: Evaluation results for KD with varying effective batch size.

kept constant at 3.4445, -4.775 and 2.0315, weight decay is 10^{-1}, $\epsilon = 10^{-7}$ and the number of Newton-Schulz iterations is constant at 5.

For all the tuning runs above, we run the evaluation datasets 5 times with varying seed 10, 20, 30, 40 and 50, and take the average of these values for each evaluation dataset. The value that is eventually plotted in Figure 6 is the checkpoint that corresponds to the highest average accuracy for a particular combination of batch size and learning rate. Note that all hyper parameter sweeps use cosine annealing scheduling.

Appendix C. Detailed results for batch size vs. accuracy

C.1. Detailed results with fp16

The detailed results for ABKD with all datasets, optmizers and batch sizes is shown in Section C.1.

Batch size	Dolly Eval			Self-Instruct			Vicuna Eval			Super-Natural			Unnatural			Average		
	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon
2	18.90	26.26	25.06	6.73	12.38	12.28	11.37	15.46	13.76	11.83	26.83	25.12	25.33	25.53	24.56	14.83	21.29	20.16
4	22.21	26.21	25.86	8.91	12.75	12.15	13.92	15.53	15.04	15.33	26.79	26.04	25.53	25.63	25.37	17.18	21.38	20.89
8	24.22	26.32	26.24	10.29	12.73	12.18	14.71	15.88	15.75	19.87	26.87	26.41	26.01	25.78	25.55	19.02	21.52	21.22
16	25.54	26.76	26.15	11.09	12.74	12.44	14.63	15.44	15.91	20.99	26.69	26.34	25.88	26.03	25.30	19.63	21.53	21.23
32	26.31	26.15	26.14	12.94	12.69	12.22	16.22	15.84	15.89	26.79	26.58	26.32	25.40	25.92	25.93	21.53	21.44	21.30
64	25.76	26.24	26.44	10.89	12.35	12.55	15.66	15.93	15.96	24.16	26.66	26.35	22.25	25.70	25.38	19.74	21.38	21.34
128	26.41	26.19	26.35	12.97	12.46	12.14	16.11	16.05	16.29	27.02	26.42	27.28	25.36	26.06	26.05	21.58	21.43	21.62
256	26.23	26.30	26.10	13.15	12.42	12.41	15.97	16.20	16.74	26.90	26.68	26.62	25.50	25.78	25.75	21.55	21.48	21.52
512	26.41	26.08	26.05	13.23	12.35	12.53	16.17	16.06	15.84	27.05	26.76	26.10	25.36	26.08	25.71	21.65	21.47	21.25
1024	26.09	26.01	9.55	12.18	12.38	6.81	16.25	16.51	12.02	26.60	26.52	7.32	25.81	25.63	10.62	21.38	21.41	9.26
2048	26.09	25.87	9.33	12.97	12.15	7.23	15.90	15.99	11.98	26.59	26.60	8.57	25.61	25.73	10.18	21.43	21.27	9.46
4096	25.44	25.63	9.11	12.67	12.26	6.08	15.57	15.44	11.27	26.39	26.35	6.96	25.65	25.57	8.91	21.14	21.05	8.47
8192	24.18	25.41	11.44	11.85	12.26	5.10	13.96	15.70	13.42	24.56	26.60	6.46	24.68	26.07	6.54	19.84	21.21	8.59

Table 3: Evaluation results for GKD with varying effective batch size.

Batch size	Dolly Eval			Self-Instruct			Vicuna Eval			Super-Natural			Unnatural			Average		
	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon
2	24.99	NA	17.01	10.98	NA	7.78	14.63	NA	9.53	22.76	NA	14.92	20.41	NA	12.77	18.75	NA	12.40
4	24.47	NA	22.62	10.82	NA	10.13	15.36	NA	13.22	21.77	NA	21.18	19.87	NA	20.29	18.46	NA	17.49
8	24.14	NA	23.56	10.23	NA	10.71	15.32	NA	14.67	20.88	NA	20.96	18.45	NA	19.20	17.80	NA	17.82
16	24.10	NA	23.93	10.74	NA	10.52	14.56	NA	14.68	21.38	NA	21.53	19.03	NA	19.80	17.96	NA	18.09
32	NA	23.98	23.78	NA	10.11	10.87	NA	15.06	14.73	NA	18.16	22.83	NA	15.79	21.13	NA	16.62	18.67
64	23.89	24.33	24.56	10.40	10.41	11.39	14.81	15.62	15.57	20.81	19.65	22.28	18.24	17.75	20.82	17.63	17.55	18.92
128	23.12	24.08	20.76	10.20	10.24	9.21	14.69	15.55	13.89	20.65	20.11	18.49	17.91	17.23	16.69	17.31	17.44	15.81
256	23.63	24.02	23.08	10.39	10.62	9.75	15.13	15.28	15.25	20.61	20.83	18.64	18.61	19.10	16.82	17.67	17.97	16.71
512	21.75	23.20	11.99	9.74	10.14	4.86	14.09	15.39	9.32	19.87	20.18	7.52	17.27	17.77	6.99	16.55	17.33	8.13
1024	21.79	22.45	14.41	9.65	9.87	5.60	14.58	14.82	12.93	19.09	19.34	8.23	16.48	16.96	8.50	16.32	16.69	9.93
2048	20.07	20.09	6.77	9.12	9.25	4.25	13.89	14.04	9.34	16.84	16.61	5.14	15.03	14.95	6.38	14.99	14.99	6.38
4096	16.38	16.37	6.16	7.76	7.91	4.25	12.20	12.24	10.19	14.10	13.57	4.13	13.03	12.83	5.74	12.69	12.58	6.10
8192	14.47	13.90	5.76	7.25	6.81	4.10	11.38	11.00	10.33	13.18	11.72	3.46	12.61	11.64	4.83	11.78	11.01	5.70

Table 4: Evaluation results for SFT with varying effective batch size.

Table 5: Best mean ROUGE-L scores for ABKD across effective batch sizes, comparing AdamW, SOAP, and Muon on Dolly Eval, Self-Instruct, Vicuna Eval, Super-Natural, Unnatural, and the overall average.

Batch size	Dolly Eval			Self-Instruct			Vicuna Eval			Super-Natural			Unnatural			Average		
	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon
512	26.38	26.26	26.34	12.61	12.33	12.26	16.29	16.56	16.49	25.24	25.43	25.22	27.34	27.61	27.32	21.57	21.64	21.53
2048	26.29	26.07	26.26	12.60	12.67	12.04	16.64	16.26	16.94	25.14	25.45	25.14	27.24	27.22	27.22	21.58	21.53	21.52
4096	–	–	26.42	–	–	11.83	–	–	16.24	–	–	25.99	–	–	27.87	–	–	21.67
8192	25.78	25.57	26.00	12.36	12.14	12.59	16.15	16.38	16.46	24.68	25.10	24.16	26.41	26.77	26.39	21.08	21.19	21.12
32768	24.84	25.09	24.75	11.88	11.79	11.36	15.62	15.48	15.73	24.87	24.74	23.83	25.34	25.18	24.48	20.51	20.46	20.03
131072	23.10	20.23	22.68	11.45	9.47	11.25	16.76	13.20	16.37	24.16	15.50	23.86	23.92	17.00	22.87	19.88	15.08	19.40

Table 6: Best mean ROUGE-L scores for GKD across effective batch sizes, comparing AdamW, SOAP, and Muon on Dolly Eval, Self-Instruct, Vicuna Eval, Super-Natural, Unnatural, and the overall average.

Batch size	Dolly Eval			Self-Instruct			Vicuna Eval			Super-Natural			Unnatural			Average		
	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon
512	25.97	25.99	25.94	12.32	11.83	12.06	16.03	16.25	16.05	25.01	23.66	25.37	26.31	24.82	26.60	21.13	20.51	21.20
2048	25.78	26.00	25.91	11.94	11.64	12.12	15.90	16.04	15.69	25.80	23.67	25.61	26.66	25.01	26.30	21.21	20.47	21.13
8192	25.29	24.99	25.44	12.23	11.05	12.07	15.18	15.52	15.25	26.04	22.43	25.64	26.08	22.56	26.20	20.96	19.31	20.92
32768	25.46	24.38	25.53	12.58	11.34	12.21	15.66	14.50	15.47	26.97	25.14	26.56	26.73	23.89	26.85	21.48	19.85	21.32
131072	24.90	20.23	24.04	12.60	9.47	12.33	13.60	13.20	13.79	27.54	15.50	25.44	26.52	17.00	24.57	21.03	15.08	20.04

C.2. Detailed results with bf16

This subsection summarizes the best mean ROUGE-L results obtained at each effective batch size for four training methods: ABKD, GKD, SFT, and KD. Each table compares AdamW, SOAP, and Muon using the best run per optimizer and batch-size setting, reported across Dolly Eval, Self-Instruct, Vicuna Eval, Super-Natural, Unnatural, and the overall average.

ABKD shows the strongest overall performance across the optimizer sweep, with SOAP and Muon generally matching or slightly exceeding AdamW at moderate batch sizes.

GKD remains competitive across most batch sizes, with Muon often giving the strongest average among the three optimizers while SOAP degrades more noticeably at the largest batch sizes.

SFT is substantially weaker than the distillation-based methods across the full sweep, and Muon is only available at the largest batch size in this summary.

KD is more stable than SFT across batch sizes, with relatively small differences among optimizers at the largest scale and modest gains from Muon at smaller and medium scales.

Table 7: Best mean ROUGE-L scores for SFT across effective batch sizes, comparing AdamW, SOAP, and Muon on Dolly Eval, Self-Instruct, Vicuna Eval, Super-Natural, Unnatural, and the overall average.

Batch size	Dolly Eval			Self-Instruct			Vicuna Eval			Super-Natural			Unnatural			Average		
	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon
512	22.50	21.42	–	10.05	9.75	–	14.84	14.08	–	18.28	17.63	–	20.17	19.86	–	17.17	16.55	–
2048	20.81	18.90	–	9.23	8.56	–	13.40	13.28	–	15.47	15.00	–	17.68	16.79	–	15.32	14.51	–
8192	16.59	14.52	–	7.31	7.11	–	12.12	11.21	–	12.68	11.96	–	13.40	13.13	–	12.42	11.58	–
32768	11.56	10.04	–	6.35	5.58	–	9.14	10.68	–	11.17	8.92	–	11.36	9.13	–	9.92	8.87	–
131072	9.70	6.36	8.55	5.22	4.59	4.83	9.20	10.57	9.53	9.14	4.48	8.00	6.67	4.69	7.15	7.99	6.14	7.61

Table 8: Best mean ROUGE-L scores for KD across effective batch sizes, comparing AdamW, SOAP, and Muon on Dolly Eval, Self-Instruct, Vicuna Eval, Super-Natural, Unnatural, and the overall average.

Batch size	Dolly Eval			Self-Instruct			Vicuna Eval			Super-Natural			Unnatural			Average		
	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon	AdamW	SOAP	Muon
512	23.46	22.66	23.59	10.14	10.38	10.48	15.35	14.17	15.27	17.37	18.02	17.91	20.22	20.20	19.82	17.31	17.09	17.41
2048	21.98	21.52	22.68	10.06	9.83	10.20	14.38	14.16	13.90	17.37	17.05	17.83	19.24	19.15	19.98	16.61	16.34	16.92
8192	20.89	20.77	21.02	9.64	9.39	10.05	13.33	13.30	13.84	16.22	16.17	16.84	17.94	18.07	18.32	15.61	15.54	16.01
32768	20.39	20.30	20.38	8.86	9.12	8.85	13.34	13.76	14.87	15.93	15.42	15.20	17.26	16.91	16.33	15.16	15.10	15.13
131072	20.20	20.23	20.02	9.34	9.47	9.00	13.16	13.20	13.86	15.53	15.50	15.52	17.09	17.00	16.89	15.06	15.08	15.06