EXPERTMUON-COMPASS: ALIGNMENT-GUIDED STEP SIZES FOR MIXTURE-OF-EXPERTS TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-experts (MoE) language models send each token to a few experts, so each expert is trained on a different part of the data, and this part changes during training. With a shared learning rate, Muon applies updates of roughly the same size to expert matrices of the same shape, even when an expert's update is poorly aligned with its current gradient. We here propose ExpertMuon-Compass (Compass), which multiplies the Muon step of each expert by two factors. A family factor compares the cosine between the expert's orthogonalized update and its gradient with the same cosine for the other experts in its layer. A scalar radius aggregates the alignment between corresponding rows of the update and gradient into one step-length multiplier. Compass keeps the update direction and the momentum buffer of Muon. In pretraining on FineWeb-Edu, Compass with Nesterov momentum on all matrices performs as well as or better than Muon, NorMuon, and other optimizers. Adding its factors to NorMuon gives the same or a lower loss than NorMuon. Compass is the most effective when the data seen by each expert varies during training, for example, as when the languages of a multilingual corpus arrive in separate blocks or in continual-learning settings. With Compass, the expert load stays balanced, and the router assigns tokens to experts more decisively. We also prove a perturbation bound for the two factors.

1 INTRODUCTION

Mixture-of-experts (MoE) language models increase capacity by activating a small subset of expert networks for each token (Shazeer et al., 2017; Fedus et al., 2022). The router also decides what an expert learns from, since the expert receives gradients only from the tokens assigned to it, and these assignments shift during training. Muon (Jordan et al., 2024) is now used to train large MoE models (Liu et al., 2025). It orthogonalizes the momentum of a weight matrix before the update, so the update has a nearly fixed norm. With a shared learning rate, the expert matrices then move by about the same amount at each step, even when an update agrees only weakly with the current gradient. Weak alignment can occur when the tokens routed to an expert change while its momentum still reflects earlier tokens.

Current MoE training handles the experts through the router and the learning rate. Load balancing adjusts the router with an auxiliary loss or with bias updates (Fedus et al., 2022; Wang et al., 2024). The learning rate is usually shared across all experts (Muennighoff et al., 2025; Liu et al., 2025), and Excitation scales an expert's update by how often it is used (Shaier, 2026). In these approaches, the step size of an expert is independent of how well its update agrees with its current gradient. We propose ExpertMuon-Compass (Compass), which sets the step of each expert based on this agreement, measured on the orthogonalized update that Muon applies. Compass computes the cosine between the update and the gradient of an expert matrix and compares it with the cosines of the other experts in the layer. An expert with a below-average cosine takes a proportionally smaller step, and the factors average to one over the experts. A second factor, the scalar radius, aggregates the alignment between corresponding rows of the update and gradient into a multiplier that can shorten the step while preserving its direction. Both factors are scalars. Compass thus keeps the update direction and the single momentum buffer of Muon, and the factors need only inner products and row norms. They can also be added to other Muon-based updates such as NorMuon (Li et al., 2025).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

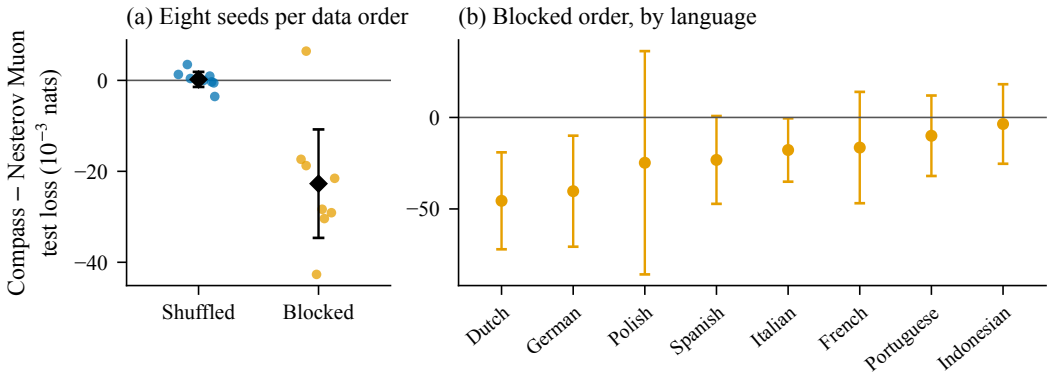


Figure 1: **Compass under language-blocked data order.** Final test loss of Compass minus Nesterov Muon on an eight-language corpus. (a) Each point is one seed; diamonds and bars show the mean and 95% confidence interval over eight seeds. (b) Mean difference and 95% confidence interval for each language under blocked order, in the same units. Compass uses Nesterov momentum for expert matrices and heavy-ball momentum for attention and router matrices, and learning rates are tuned separately for each method and data order (Section 4.4).

Changing the data that each expert sees during training gives a direct test of when expert step sizes matter under distribution shift (as occurs, e.g., in continual learning). Hence, we train an eight-language MoE model on the same token batches in two orders, once shuffled and once grouped into repeated single-language blocks. With shuffled batches, Compass and Nesterov Muon reach nearly the same test loss. The blocked order changes the data seen by the experts at each language switch, and here Compass reaches a lower test loss than Nesterov Muon in all eight languages (Figure 1). Adding the Compass factors to the update of NorMuon also lowers its test loss in every language.

Our contributions are as follows. (1) Compass, a step-size rule for the expert matrices of Muon that uses the cosine between each expert's orthogonalized update and its gradient and keeps the update direction. (2) Pretraining experiments on FineWeb-Edu with training budgets from 8.2M to 1.05B tokens. At the shortest budget, we compare Compass with Nesterov Muon, NorMuon, heavy-ball Muon, Excitation, Magma, and AdamW (Jordan et al., 2024; Li et al., 2025; Shaier, 2026; Joo et al., 2026; Loshchilov & Hutter, 2019); at longer budgets, we compare it with Nesterov Muon and NorMuon. We tune the learning rate of every method in the same way, and our ablations remove each factor or replace both factors with a constant expert scale. With Nesterov momentum on all matrices, Compass performs as well as or better than these optimizers, and adding the Compass factors to NorMuon gives the same or a lower loss than NorMuon. (3) An analysis of training. Compass keeps the expert load balanced and makes the router assign tokens to experts more decisively. In the language-blocked experiment, its factors spread out earliest in each block, soon after the data seen by the experts change. (4) A bound on how much the two factors can change the update when the gradients change.

2 Related work

We review matrix optimizers, MoE training, and work on alignment and expert updates.

Matrix optimizers. Muon orthogonalizes the momentum of each weight matrix with a Newton-Schulz iteration (Jordan et al., 2024). Its reference implementation uses Nesterov momentum (Jordan et al., 2024), and in large models its behavior depends on how the update is scaled, how parameters are grouped, and the numerical precision (Liu et al., 2025; Khona et al., 2026). NorMuon adds row-wise second-moment normalization to the orthogonalized update (Li et al., 2025). SOAP runs Adam in the eigenbasis of Shampoo's preconditioner (Vyas et al., 2025). Compass keeps the single momentum buffer of Muon and adds scalar factors for expert matrices.

Training mixture-of-experts models. Work on training MoE models has focused mostly on the router. MoE training usually balances expert load with an auxiliary loss (Fedus et al., 2022) or with router bias updates (Wang et al., 2024), and our models use the bias updates. OLMoE is trained with AdamW (Muennighoff et al., 2025), and Moonlight trains an MoE model with Muon (Liu et al., 2025). Load-balancing methods act on the router, and OLMoE and Moonlight use one learning rate for all experts. Compass sets the step of each expert from its own update and gradient.

Alignment and expert updates. Alignment between an update and the gradient has also been used to modify the update. Cautious optimizers use the sign alignment between an update and the gradient to mask individual coordinates (Liang et al., 2026). Magma combines block-level alignment with stochastic masking (Joo et al., 2026). Excitation sets expert update multipliers from expert utilization (Shaier, 2026). OMoE encourages diverse expert representations through orthogonal optimization (Liu et al., 2024). Compass measures alignment after orthogonalization and compares an expert with the matching matrices of the other experts in its layer. It also combines the row cosines of a matrix into a single scalar, preserving the direction computed by Muon.

3 EXPERTMUON-COMPASS

Compass keeps the Muon direction of each expert matrix and sets its length using two factors computed from alignment, and Section 3.4 bounds how much these factors can change the update.

3.1 UPDATE DIRECTION AND PARAMETER GROUPS

We group the matching projection matrices of all experts in one layer into a *family*; for example, the first projection of every expert in a layer forms one family. Expert e has weights $W_{e,t} \in \mathbb{R}^{m\times n}$ and clipped gradient $G_{e,t}$, and rows index the output dimension. With $\beta = 0.95$ and $M_{e,0} = 0$, the momentum and the Nesterov input are

$$M_{e,t} = \beta M_{e,t-1} + G_{e,t}, \qquad D_{e,t} = G_{e,t} + \beta M_{e,t}. \tag{1}$$

The Nesterov input uses the updated buffer. Five Newton-Schulz iterations with the coefficients of Jordan et al. (2024) give the orthogonalized direction $O_{e,t} = \mathcal{N}_5(D_{e,t})$. In exact arithmetic, they keep the singular vectors of $D_{e,t}$ and bring its singular values closer together (Appendix C.2). As in Liu et al. (2025), we multiply this direction by the shape factor $s = 0.2\sqrt{\max(m, n)}$, which makes the size of the update close to that of an AdamW update, and write $U_{e,t} = sO_{e,t}$.

Compass uses the Nesterov input for expert matrices. For attention and router matrices, it orthogonalizes either the heavy-ball momentum M_t (written N/H) or the Nesterov input $G_t + \beta M_t$ (written N/N); in both cases, the family factor c and the scalar radius α defined below are the same. The N/N configuration matches the momentum convention of the Nesterov Muon baseline. We use AdamW for embeddings, the untied output head, biases, and normalization parameters.

3.2 EXPERT AND ROW SCALING

We drop the step index t of the matrices and scores in this subsection. For arrays of equal shape, define

$$\cos(A, B) = \frac{\langle A, B\rangle_F}{\|A\|_F \|B\|_F}. \tag{2}$$

We measure alignment on the unscaled direction O_e. The score of expert e is $a_e = [\cos(O_e, G_e)]_+$, where $[x]_+ = \max(0, x)$, and a bar denotes the mean over the family. To avoid division by zero, we add small positive constants to the scores and denominators (Appendix C). The family factor is

$$u_e = a_e/\overline{a}, \qquad v_e = \min(1, u_e), \qquad c_e = v_e/\overline{v}. \tag{3}$$

The first normalization compares an expert with its family, and the cap limits this ratio at one. The second normalization makes the family multipliers average to one, setting their relative scale without fixing the total norm of the expert updates. Capping gives equal preliminary weights to above-average alignment scores, limiting the influence of unusually large scores on relative expert step sizes.

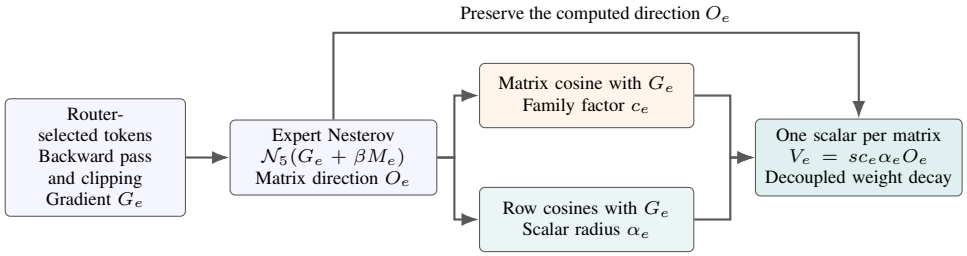


Figure 2: **The Compass update for one expert matrix.** Alignment is measured on the orthogonalized direction O_e. The family factor c_e compares whole matrices within a family, and the scalar radius α_e summarizes row-level alignment. Both scale the same direction. Router load balancing is handled separately by bias updates.

The second factor works on the rows of each expert matrix. Each row r receives the score $b_{er} = [\cos(O_{er}, G_{er})]_+$ in the same way. With $\bar{b}_e$ the mean over the rows of expert e, define

$$q_{er} = \min(1, b_{er}/\bar{b}_e), \qquad h_{er} = 1 + \lambda_t(q_{er} - 1), \tag{4}$$

where $\lambda_t = \eta_t/\eta_{\max}$ is the current learning rate divided by its peak value, so the row factors move toward one as the learning rate decays. The scalar radius is the norm of the row-weighted matrix divided by the norm of O_e,

$$\alpha_e = \sqrt{\frac{\sum_r h_{er}^2 \|O_{er}\|_2^2}{\|O_e\|_F^2}}, \qquad V_e = c_e\alpha_e U_e, \tag{5}$$

so that

$$\|\alpha_e O_e\|_F = \|\mathrm{diag}(h_e) O_e\|_F. \tag{6}$$

Weighting the rows directly would change the direction of O_e. The scalar α_e keeps the direction and takes only the length of the row-weighted matrix. It lies between zero and one. We call $c_e\alpha_e$ the expert scale, and it can exceed one because c_e can. Figure 2 shows the whole update for one expert matrix.

3.3 Complete update

With learning rate η_t and decoupled weight decay ω, each expert matrix is updated as

$$W_{e,t+1} = (1 - \eta_t\omega)W_{e,t} - \eta_t V_{e,t}. \tag{7}$$

Experts that receive zero tokens in a step are excluded from the family means and keep their weights and momentum. Algorithm 1 in Appendix A summarizes the update, and Appendix C gives the constants and the FP32 order of operations.

3.4 Perturbation bound for the two factors

Both factors divide a score by a mean score, and a small mean can amplify errors in the scores. We therefore bound how much the update at one step changes when the directions and the scores change, for example, when the gradients come from a different batch. We drop the step index and assume exact arithmetic, with the small positive constants of Appendix C added to the scores. The family factor of Equation (3) equals $c_e = \min(a_e, \bar{a})/\mu$, where $\mu = \bar{a}\,\bar{v}$ is the family mean of the capped scores $\min(a_e, \bar{a})$, and the row factors divide the row scores by their means $\nu_e = \bar{b}_e$. A tilde marks the quantities computed from a second family of directions and gradients.

Theorem 1 (Perturbation bound for the family factor and radius)**.** *Let V_e and $\widetilde{V}_e$ be the updates of Equation* (5) *for a family of E matrices, computed from (O_e, G_e) and $(\widetilde{O}_e, \widetilde{G}_e)$ with the same shape factor s and schedule multiplier $\lambda \in [0, 1]$. Let*

$$\delta_O = \max_e \|O_e - \widetilde{O}_e\|_F, \qquad \delta_a = \max_e |a_e - \widetilde{a}_e|, \qquad \delta_b = \max_{e,r} |b_{er} - \widetilde{b}_{er}|, \qquad R = \max_e \|\widetilde{O}_e\|_F,$$

$A = \min\{\mu, \widetilde{\mu}\}$ *and* $B = \min_e \min\{\nu_e, \widetilde{\nu}_e\}$. *Then* $A, B > 0$ *and*

$$\frac{1}{E}\sum_e \|V_e - \widetilde{V}_e\|_F \le s\left(3\delta_O + \frac{2R\delta_a}{A} + \frac{2\lambda R\delta_b}{B}\right). \tag{8}$$

Because A is the smaller of two means over the experts, a few experts with scores near zero have a small effect on it. Similarly, B is the smallest of the row means ν_e and $\widetilde{\nu}_e$, and a few rows with scores near zero have a small effect on each of these means. The row term is proportional to λ, so it vanishes at the end of training, when the row factors equal one. The bound also holds with the constants of Appendix C, for zero matrices and for scores at the caps, and it allows the scores to depend on the same gradient as the direction. We prove it in Appendix G, where we also convert gradient errors into score errors, and Appendix C.5 compares the scalar radius with row weighting at equal step length. Kim & hwan Oh (2026) and Shulgin et al. (2025) analyze finite Newton-Schulz iterations and inexact Muon updates; our analysis covers the family normalization and the scalar radius.

4 Experiments

We compare Compass with the baselines at 8.2M tokens (Section 4.2), in longer training where weight decay matters (Section 4.3), and under a language-blocked data order (Section 4.4).

4.1 Experimental setup

We pretrain MoE language models from random initialization on FineWeb-Edu and FineWeb2 (Penedo et al., 2024; 2025). The models have six pre-norm Transformer layers, GELU experts, and top-2 routing, and expert load is balanced with auxiliary-loss-free router bias updates (Wang et al., 2024). We keep parameters, gradients, optimizer states, and matrix products in FP32 for every method, so that the comparison reflects the optimizer updates.

On FineWeb-Edu, we use widths 1,024 and 2,048 with eight experts per layer (329.8M and 1,112.5M total parameters). The short runs train for 2,000 steps with batch size 16 and context length 256, for 8.2M tokens. The long runs use width 1,024, context length 1,024, and 4,000 steps (65.5M tokens, 330.6M parameters), and we extend them to 16,000 and 64,000 steps (262M and 1.05B tokens). Training sequences are sampled from a 57M-token training split for runs up to 65.5M tokens and from 380M- and 1.14B-token splits for the two longer budgets.

The matrix and AdamW learning rates follow the cosine schedule

$$\eta_t = \eta_{\max}\lambda_t, \qquad \lambda_t = \tfrac{1}{2}\{1 + \cos(\pi(t-1)/(T-1))\}, \qquad t = 1, \dots, T, \tag{9}$$

with zero warmup. In the runs from random initialization, the parameters that the Muon-based methods update with AdamW use a peak learning rate of 0.003, $(\beta_1, \beta_2) = (0.9, 0.95)$, and weight decay of 0.01. Unless stated otherwise, the matrix weight decay is $\omega = 0.01$.

For each method, we tune the learning rate over three values on two tuning seeds and select the value with the lowest mean final validation loss. We then train eight evaluation seeds at the selected rate and report their mean final validation loss. All methods use the same random seeds, which set both the initialization and the order of the training batches. We therefore compare the methods seed by seed, with 95% confidence intervals (CIs) from paired t-tests (Appendix B).

Baselines and ablations. Nesterov Muon orthogonalizes $G + 0.95M$ for all hidden matrices, and heavy-ball Muon orthogonalizes M. They share the shape factor and AdamW settings of Compass. For NorMuon, we use the reference implementation of Li et al. (2025), with its own row normalization and update scaling, in FP32. We implemented Excitation from its published update rule on top of heavy-ball Muon. For Magma, we use a public AdamW-based implementation. The AdamW baseline updates all parameters with its own tuned learning rate. The ablations are expert-only Nesterov (Compass with $c = \alpha = 1$); a constant expert scale, which multiplies every expert step by the mean of $c\alpha$ from a separate Compass run and tunes only its learning rate; and Compass with only the family factor or only the radius. The constant-scale control tests whether alignment-dependent variation in expert step sizes improves on a fixed multiplier matched to their average scale. Computing

Table 1: Pretraining on FineWeb-Edu at 8.2M tokens (context length 256). Final validation loss (nats), mean ± standard deviation over eight seeds. The momentum column gives the choice for expert matrices and for attention and router matrices (N: Nesterov, H: heavy-ball); Adam marks the AdamW-based methods. The best mean in each column is in bold.

Method	Momentum	Width 1,024	Width 2,048
Compass	N/H	5.45397 ± 0.00710	5.32207 ± 0.01184
Compass	N/N	$\mathbf{5.44908} \pm 0.00768$	5.32340 ± 0.01162
Nesterov Muon	N/N	5.45252 ± 0.00861	5.32296 ± 0.00973
NorMuon	N/N	5.45211 ± 0.00619	5.32626 ± 0.00909
Heavy-ball Muon	H/H	5.46876 ± 0.00673	5.32991 ± 0.01169
Excitation	H/H	5.46430 ± 0.00579	5.32789 ± 0.01066
Magma	Adam	5.85811 ± 0.01501	5.82693 ± 0.04095
AdamW	Adam	6.03952 ± 0.01077	5.93637 ± 0.02066
Ablations of Compass (N/H)			
Expert-only Nesterov	N/H	5.46000 ± 0.00322	5.32783 ± 0.01330
Constant expert scale	N/H	5.46146 ± 0.00635	5.32505 ± 0.01347
Family factor only	N/H	5.45695 ± 0.00797	5.32358 ± 0.01078
Radius only	N/H	5.45528 ± 0.00548	$\mathbf{5.32186} \pm 0.01198$

the Compass factors adds 4 to 7% to the wall-clock time of a training run (Table 6 in Appendix A). Learning-rate grids and the remaining settings are in Appendix A, and Appendix E applies Compass to JetMoE-8B, a released MoE model with about 8B parameters (Shen et al., 2024).

4.2 Pretraining at 8.2M tokens

Table 1 summarizes the short runs, and Figure 5 and Table 9 in Appendix B show the paired differences. With Nesterov momentum on all matrices (N/N), Compass has the lowest mean loss at width 1,024, below NorMuon on six of eight seeds (paired permutation test $p = 0.031$). At width 2,048, it has a lower mean loss than NorMuon and matches Nesterov Muon. With heavy-ball momentum on attention and router matrices (N/H), Compass has a lower loss than heavy-ball Muon on all 16 seeds. An earlier comparison with heavy-ball Muon at these widths, with separate learning-rate sweeps, is in Appendix D. Excitation, Magma, and AdamW have higher losses than either Compass configuration. At this budget, Compass matches or improves on each baseline.

The ablations test whether this comes from the expert factors (Table 8 in Appendix B). With the momentum settings held fixed, Compass has a lower loss than the constant expert scale at width 1,024 on all eight seeds. Compass also has a lower mean loss than expert-only Nesterov at both widths. At width 1,024 it is below the family-factor-only ablation on all eight seeds, and at width 2,048 it matches the radius-only ablation. At width 1,024, alignment-dependent expert steps thus improve on the constant expert scale; Appendix F gives more ablations up to width 2,560.

4.3 Longer training at context length 1,024

Longer training makes the weight decay more important in the comparison with NorMuon. With its selected learning rate of 0.016 and weight decay 0.01, NorMuon shrinks each weight by a factor of $1 - 1.6 \times 10^{-4}$ per step at the peak learning rate. This per-step decay is 6.4 to 7.1 times that of Compass and Nesterov Muon at their selected rates. We therefore also train Compass and Nesterov Muon with the same per-step decay as NorMuon, setting ω so that $\eta_t\omega$ matches, and tune both over the rates $\{0.00125, 0.00175, 0.0025\}$. For Compass, we also evaluate this decay twice.

In Table 2, at the default decay, Nesterov momentum on the attention and router matrices lowers the loss of Compass, which then matches Nesterov Muon, and the two again reach nearly the same loss with NorMuon's per-step decay. With twice that decay, Compass has the lowest mean loss in the table, also when NorMuon and Nesterov Muon use twice the decay with the same tuning. On more seeds at twice the decay, chosen before training, Compass stays below NorMuon (lower on 22 of 32 seeds) and slightly below Nesterov Muon (30 of 46), and Compass on the NorMuon update stays below NorMuon (20 of 32). On the full 3M-token validation split, all three differences are

Table 2: Pretraining at context length 1,024 (width 1,024, 65.5M tokens). Final validation loss (nats), mean ± standard deviation over eight seeds, and the selected learning rate. The upper block uses weight decay $\omega = 0.01$. The lower block gives every method the per-step decay of NorMuon (learning rate 0.016, $\omega = 0.01$), $\eta_t\omega = 1.6 \times 10^{-4}\lambda_t$; the rows marked twice the decay use twice this per-step decay. The best mean is in bold.

Method	Momentum	Validation loss	Learning rate
Compass	N/H	4.35013 ± 0.00837	0.0025
Expert-only Nesterov	N/H	4.35145 ± 0.00875	0.0025
Constant expert scale	N/H	4.34941 ± 0.00849	0.0025
Compass	N/N	4.31807 ± 0.00516	0.0025
Nesterov Muon	N/N	4.31872 ± 0.00451	0.00225
NorMuon's per-step weight decay			
NorMuon	N/N	4.30685 ± 0.00514	0.016
Nesterov Muon	N/N	4.30257 ± 0.00850	0.00175
Compass on the NorMuon update	N/N	4.30332 ± 0.00505	0.016
Compass	N/N	4.30225 ± 0.00407	0.00175
NorMuon, twice the decay	N/N	4.29312 ± 0.00762	0.016
Nesterov Muon, twice the decay	N/N	4.28791 ± 0.00760	0.00175
Compass, twice the decay	N/N	$\mathbf{4.28522} \pm 0.00449$	0.00175

Table 3: Compass against NorMuon as the training budget grows (width 1,024, context length 1,024). The decay column gives Compass's per-step weight decay as a multiple of NorMuon's; NorMuon keeps its own decay throughout. Losses are final validation losses (nats), averaged over the listed number of seeds; Δ is Compass minus NorMuon on the same seeds with its 95% confidence interval, and wins counts the seeds on which Compass has the lower loss.

Tokens	Decay	Seeds	NorMuon	Compass	Δ [95% CI]	Wins
Compass (N/N)						
65.5M	1×	16	4.30562	4.29974	$-0.00587\ [-0.01024, -0.00151]$	13/16
65.5M	2×	16	4.30562	4.28556	$-0.02005\ [-0.02384, -0.01627]$	16/16
Compass on the NorMuon update						
65.5M	1×	16	4.30562	4.30110	$-0.00452\ [-0.00727, -0.00176]$	14/16
262M	1×	8	3.70793	3.70774	$-0.00019\ [-0.00412, 0.00375]$	3/8
1.05B	1×	4	3.53628	3.52745	$-0.00882\ [-0.03393, 0.01628]$	2/4

significant, and Compass has the lowest WikiText-2 test loss of the models we scored (Appendix B). With matched weight decay, Compass therefore matches or improves on each baseline at this budget.

Table 3 extends the comparison with NorMuon to 16 seeds at 65.5M tokens and to budgets of 262M and 1.05B tokens, with each run's cosine schedule set to its full length. These extensions evaluate how the selected learning rates and weight decay transfer to longer training budgets. Compass on the NorMuon update uses NorMuon's learning rate and weight decay, so the two methods differ only in the expert factors, and we trained both at every budget on the same seeds. At 65.5M tokens, Compass has the lower loss, and the gap grows during training (Figure 4 in Appendix A). At 262M and 1.05B tokens, the two methods are close, also on 16.8M held-out tokens and on WikiText-2 (Appendix A.2). Compass (N/N) has a lower mean loss than NorMuon at both decays, and with twice the decay it is below NorMuon at every evaluation during training; part of this gap comes from the decay itself, because NorMuon with twice its decay is also 0.01254 nats below NorMuon at its default decay on the same seeds. Under a second decay rule chosen before training, which holds the total decay $\sum_t \eta_t\omega$ constant across budgets, Compass (N/N) has the lowest mean loss of the four methods at 262M tokens over nine seeds (Appendix A.2). Added to NorMuon, the factors therefore lower its loss to 65.5M tokens and give a similar loss at the longer budgets.

4.4 Language-blocked data order

FineWeb-Edu shuffles its training data in random order. To test the setting of our motivating example, in which the data seen by the experts change during training as in curriculum or continual

Table 4: Held-out FineWeb2 test loss (nats) after 16.4M training tokens, mean ± standard deviation over eight seeds. The same batches are shuffled or arranged in 100-step language blocks. Compass uses the N/H configuration, and the lower block applies the Compass factors to the expert matrices of the NorMuon update, with the learning-rate grid and weight decay of NorMuon. With shuffled batches all methods lie within 0.005 nats of each other, and the best mean under blocked order is in bold.

Method	Shuffled	Blocked
Compass (N/H)	4.40861 ± 0.00188	5.46396 ± 0.04741
Nesterov Muon	4.40840 ± 0.00249	5.48667 ± 0.04560
Heavy-ball Muon	4.40923 ± 0.00207	5.47008 ± 0.05157
Expert-only Nesterov	4.40961 ± 0.00213	5.47661 ± 0.04251
Constant expert scale	4.40866 ± 0.00380	5.46575 ± 0.04341
NorMuon	4.40463 ± 0.00202	5.46953 ± 0.04158
Compass on the NorMuon update	4.40569 ± 0.00362	$\mathbf{5.44608} \pm 0.04932$

learning, we also train on a multilingual corpus in two data orders. The FineWeb2 model has 64 experts per layer (461.2M total parameters) and trains for 4,000 steps with four sequences of length 1,024 per step (16.4M tokens). Eight languages contribute equal numbers of tokens. In the blocked order, each seed cycles five times through eight blocks of 100 single-language steps. The shuffled order uses the same 4,000 batches in random order, and both orders share the initialization and the held-out data. We tune learning rates separately for each method and order. Compass uses the N/H configuration, and we also add its factors to the NorMuon update. Test losses use 65,536 held-out tokens per language.

Under blocked order, Compass has a lower mean test loss than Nesterov Muon, heavy-ball Muon, NorMuon, expert-only Nesterov, and the constant expert scale (Table 4). Against Nesterov Muon, it is lower on seven of eight seeds (paired permutation test $p = 0.016$) and in all eight languages (Figure 1b). We repeated this comparison on eight new seeds with every setting fixed before training and found a similar gap (Table 12); over the 16 seeds, Compass is lower on 14 ($p = 0.0012$). With shuffled batches, the two methods reach nearly the same loss, and the gap under blocked order is larger than under shuffled order ($p = 0.016$). The effect of Compass therefore depends here on the data order. The constant expert scale also comes close to Compass under blocked order. In the short runs at width 1,024, Compass improves on it on all eight seeds.

Added to the NorMuon update, the Compass factors give the lowest test loss under blocked order (Table 4). This variant is below NorMuon on six of eight seeds and in all eight languages. A replication on eight new seeds gives a similar gap, and over all 16 seeds it is lower on 13 ($p = 0.0005$). With shuffled batches, it matches NorMuon (Table 13). The two methods share the update, weight decay, learning-rate grid, and tuning seeds, and each uses its own selected learning rate.

5 Analysis of training

We now examine expert scales, load balance and routing, response to language switches, and training stability with Compass. Figure 3 uses statistics recorded during the 65.5M-token runs with twice NorMuon's per-step decay.

Compass factors. The expert scale $c\alpha$ differs across the 96 expert matrices throughout training (Figure 3a). After the first 100 steps, the smallest factor stays between about 0.55 and 0.7 and the largest between 1.05 and 1.10. The mean rises from about 0.93 to one as the radius approaches one, and late in training some experts still take steps about 30% shorter than the average.

Load balance and routing. Moving step size between experts could change the routing. All four optimizers keep the expert load balanced (Figure 3b): for every method, MaxVio (Wang et al., 2024), the relative excess load of the most used expert averaged over the MoE layers, falls from about 0.2 at the first evaluation to about 0.05 at the end of training. Every expert receives tokens in all 165 runs of these methods at 65.5M tokens, at NorMuon's per-step decay and at twice that decay. The router distribution over experts is more concentrated with Compass (Figure 3c). Its mean entropy,

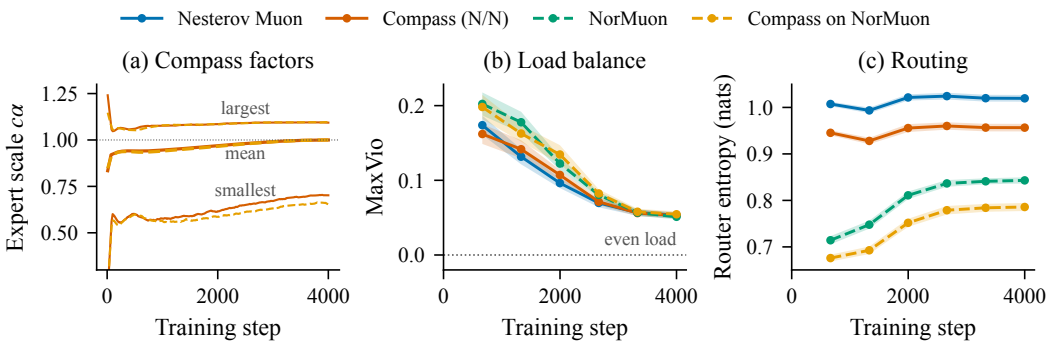


Figure 3: **Training analysis at 65.5M tokens.** All methods use twice NorMuon's per-step weight decay (33 seeds for Compass (N/N), 32 for Nesterov Muon, and 30 each for NorMuon and Compass on the NorMuon update). (a) Smallest, mean, and largest expert scale $c\alpha$ over the 96 expert matrices at each step, averaged over seeds and smoothed over 100 steps. (b) MaxVio; zero means an even load. (c) Mean entropy of the router softmax over all eight experts, computed without the router bias. Panels (b) and (c) use 32,768 fixed training tokens and average over the MoE layers; bands show 95% confidence intervals.

which Excitation also reports (Shaier, 2026), is lower with Compass than with the same update alone across seeds. It is 0.957 against 1.020 nats with the Muon update (32 seeds, $t = -16.0$, $p < 10^{-15}$) and 0.786 against 0.843 nats with the NorMuon update (30 seeds, $t = -12.4$, $p < 10^{-12}$). The gap persists at the longer budgets of Table 3, where the losses of the two methods are close. Compass on the NorMuon update has the lower router entropy on all eight seeds at 262M tokens ($t = -7.2$, $p = 0.0002$) and on three of the four seeds at 1.05B tokens. Both runs update the router the same way, so the difference arises from the expert updates; at the same load balance, tokens are assigned more clearly to particular experts.

Language switches. The factors also respond when the data seen by the experts change. In the multilingual runs of Section 4.4, we recorded the spread of $c\alpha$ across the expert matrices of Compass at NorMuon update 25, 50, 75, and 100 steps into each block. Twenty-five steps after a language switch, the standard deviation of $c\alpha$ is 0.145 against 0.136 later in the block, and its minimum is 0.26 against 0.30, on all eight seeds ($t = 12.3$ and $t = -4.5$). With shuffled batches, the spread at the same steps stays within 0.003 of its later value. Compass therefore shortens the steps of some experts most strongly early in each block, and it lowers the test loss most under blocked order.

Stability. Moving step size between experts could also make training less stable. None of these 165 runs diverged or produced a non-finite loss. Following OLMo et al. (2025), we count a loss spike when the training loss at a step is at least seven standard deviations away from the mean of the previous 1,000 steps. No run has such a spike, even with a 100-step window.

6 Conclusion

ExpertMuon-Compass sets each expert matrix step based on the agreement between its orthogonalized update and its current gradient, while keeping the direction computed by Muon. In pretraining on FineWeb-Edu, Compass with Nesterov momentum on all matrices performs as well as or better than Muon, NorMuon, and the other optimizers we tested. Added to NorMuon, Compass lowers its loss at 65.5M tokens while yielding similar loss at 262M and 1.05B tokens. In these runs, the expert load stays balanced, and the router assigns tokens to experts more decisively than with the same update alone. Compass is most useful when the data seen by the experts change, as with language blocks, reducing steps more strongly early in the block. These results suggest that expert step sizes matter more when the data of each expert change during training, as in continual pretraining or with data mixtures that vary over time.

REPRODUCIBILITY STATEMENT

The supplementary material contains our code and per-seed results. Section 3 and Appendix C define the update, including its constants and order of operations, and Section 4 describes the experiments. Appendix A lists the architectures, data splits, token counts, learning-rate grids, selection rule, and baseline settings, and Appendix B describes the statistical tests. Appendix E gives the JetMoE settings and results, and Appendix G contains the proofs.

ETHICS STATEMENT

This work uses publicly released language models and datasets, and their usage conditions apply to the trained checkpoints and derived data. We keep training, model selection, and held-out evaluation data separate. Models trained on these corpora can reproduce biases and harmful content in the data, so any deployment needs a separate evaluation.

AI USE STATEMENT

Generative AI tools assisted with research planning, experimental design, implementation, statistical analysis, the proofs, and the preparation of text and figures. The authors checked the code, proofs, calculations, and claims against the experimental outputs; all reported measurements come from completed training and evaluation runs, and the authors take full responsibility for the content.

REFERENCES

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

Nicholas J. Higham. *Functions of Matrices*. Society for Industrial and Applied Mathematics, 2008. doi: 10.1137/1.9780898717778. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898717778.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, second edition, 2013.

Taejong Joo, Wenhan Xia, Cheolmin Kim, Ming Zhang, and Eugene Ie. On surprising effectiveness of masking updates in adaptive optimizers, 2026. URL https://arxiv.org/abs/2602.15322.

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/.

Mikail Khona, Aditya Vavre, Boxiang Wang, Deyu Fu, Hao Wu, Mike Chrzanowski, Bryan Catanzaro, Dheevatsa Mudigere, Jeff Pool, Michael Lightstone, Mohammad Shoeybi, Mostofa Patwary, Nima Tajbakhsh, and Tijmen Blankevoort. Soap, muon, and beyond: Pushing llm pretraining scales, 2026. URL https://arxiv.org/abs/2607.20548.

Gyu Yeol Kim and Min hwan Oh. Convergence of muon with newton-schulz, 2026. URL https://arxiv.org/abs/2601.19156.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer,

Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: in search of the next generation of training sets for language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Zichong Li, Liming Liu, Chen Liang, Weizhu Chen, and Tuo Zhao. Normuon: Making muon more efficient and scalable, 2025. URL https://arxiv.org/abs/2510.05491.

Kaizhao Liang, Lizhang Chen, Bo Liu, and Qiang Liu. Cautious optimizers: Improving training with one line of code, 2026. URL https://arxiv.org/abs/2411.16085.

Boan Liu, Liang Ding, Li Shen, Keqin Peng, Yu Cao, Dazhao Cheng, and Dacheng Tao. Diversifying the mixture-of-experts representation for language models with orthogonal optimizer, 2024. URL https://arxiv.org/abs/2310.09762.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for llm training, 2025. URL https://arxiv.org/abs/2502.16982.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Evan Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. OLMoe: Open mixture-of-experts language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xXTkbTBmqq.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 olmo 2 furious, 2025. URL https://arxiv.org/abs/2501.00656.

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, 2024. URL https://arxiv.org/abs/2406.17557.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. Fineweb2: One pipeline to scale them all – adapting pre-training data processing to every language, 2025. URL https://arxiv.org/abs/2506.20920.

Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, third edition, 1976.

Sagi Shaier. Excitation: Momentum for experts, 2026. URL https://arxiv.org/abs/2602.21798.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017. URL https://arxiv.org/abs/1701.06538.

Yikang Shen, Zhen Guo, Tianle Cai, and Zengyi Qin. Jetmoe: Reaching llama2 performance with 0.1m dollars, 2024. URL https://arxiv.org/abs/2404.07413.

Egor Shulgin, Sultan AlRashed, Francesco Orabona, and Peter Richtárik. Beyond the ideal: Analyzing the inexact muon update, 2025. URL https://arxiv.org/abs/2510.19933.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. Soap: Improving and stabilizing shampoo using adam, 2025. URL https://arxiv.org/abs/2409.11321.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts, 2024. URL https://arxiv.org/abs/2408.15664.

A EXPERIMENTAL DETAILS FOR THE PRETRAINING RUNS

This appendix gives the architectures and training settings, additional runs at 262M and 1.05B tokens, the data splits, and the learning-rate selection and baselines.

A.1 ARCHITECTURES AND TRAINING

All pretraining models are decoder-only Transformers with six pre-norm blocks, learned positional embeddings, full causal attention, and a final LayerNorm. Table 5 gives their sizes. Each expert has two linear projections with a GELU between them; the embedding and output layers are untied, and we use the GPT-2 tokenizer (vocabulary size 50,257).

Table 5: Model sizes. In the FineWeb-Edu models, the expert hidden width is twice the model width. The multilingual model has about 71.08M active parameters per token.

	FineWeb-Edu		
Setting	8.2M tokens	65.5M to 1.05B tokens	Multilingual
Model width	1,024 / 2,048	1,024	512
Attention heads	16	16	8
Experts per layer	8	8	64
Experts per token	2	2	2
Expert hidden width	2,048 / 4,096	2,048	1,024
Context length	256	1,024	1,024
Total parameters	329,781,248 / 1,112,547,328	330,567,680	461,154,304

The router takes a softmax over all experts, keeps the two with the highest score plus bias, and renormalizes their probabilities. After each step, we update the bias of expert e as $\xi_e \leftarrow \xi_e + 0.02\,\mathrm{sign}(1/E - \ell_e)$, where ℓ_e is its share of the layer's routed tokens; this rule is the same for all optimizers, and evaluation uses the final biases. We clip gradients to a global norm of one and run all computation in FP32, without automatic mixed precision or TF32.

Algorithm 1 One training step of Compass

Require: Clipped gradients, weights, momentum buffers and learning rates at step t
1: Apply the AdamW update to embeddings, output head, biases, and normalization parameters
2: **for** each attention or router matrix with a gradient **do**
3: $M \leftarrow 0.95M + G$, $O \leftarrow \mathcal{N}_5(M)$
4: Apply weight decay and add $-\eta_t s O$ to the weights
5: **end for**
6: **for** each expert family **do**
7: Include every matrix that has a gradient, including zero gradients
8: **for** each included matrix e **do**
9: $M_e \leftarrow 0.95M_e + G_e$, $O_e \leftarrow \mathcal{N}_5(G_e + 0.95M_e)$
10: Compute the alignment score a_e from O_e and G_e
11: **end for**
12: Compute c_e with Equation (3)
13: **for** each included matrix e **do**
14: Compute the row factors and α_e with Equations (4) and (5)
15: Apply weight decay and add $-(\eta_t c_e) s(\alpha_e O_e)$ to W_e
16: **end for**
17: **end for**

For Compass (N/N), line 3 of Algorithm 1 uses $\mathcal{N}_5(G + 0.95M)$ in place of $\mathcal{N}_5(M)$, and the rest of the algorithm is unchanged. Its data, architecture, schedule, and seeds match the other runs at each budget, and Table 7 gives its learning-rate sweeps.

For the runs with NorMuon's per-step decay, we set $\omega = 1.6 \times 10^{-4}/\eta_{\max}$, so that $\eta_t\omega = 1.6 \times 10^{-4}\lambda_t$ at every step. Table 7 gives the learning-rate grids and the selected rates of these runs;

among the Compass settings, twice this decay at 0.00175 has the lowest mean tuning loss. Compass on the NorMuon update applies the Compass factors to the expert matrices of the NorMuon update from the reference implementation, with NorMuon's own weight decay, and measures its cosines on the NorMuon update itself. As in NorMuon, an expert without a gradient in a step receives a zero gradient and still takes an update from its momentum, without weight decay or Compass factors.

Each FineWeb-Edu run uses one NVIDIA RTX PRO 6000 Blackwell GPU (PyTorch 2.11, CUDA 12.8), except the 262M- and 1.05B-token pairs of Table 3, which use one GH200 GPU per run (PyTorch 2.12, CUDA 13.2) with the same code, settings, and tokenized data. The two methods of a seed always run on the same GPU type. Eleven NorMuon runs at 262M and 1.05B tokens, repeated on different nodes, reproduced their final validation losses bit for bit. The multilingual runs use GH200 GPUs and run several independent runs on one GPU, which yields the same parameters, optimizer states, and losses as running alone. Table 6 gives the wall-clock cost of the Compass factors.

Table 6: Wall-clock cost of the Compass factors. Each pair uses the same seed on the same machine, and the time covers the whole run, including validation. The last column gives the extra time of Compass relative to the method without the factors.

Tokens	GPU	Comparison	Pairs	Extra time
65.5M	RTX PRO 6000	Compass (N/N) and Nesterov Muon, twice the decay	4	4.3%
65.5M	RTX PRO 6000	Compass on the NorMuon update and NorMuon	5	3.8%
262M	GH200	Compass on the NorMuon update and NorMuon	8	5.2%
1.05B	GH200	Compass on the NorMuon update and NorMuon	4	6.6%

A.2 Additional runs at 262M and 1.05B tokens

More seeds. We also trained long-budget pairs on the RTX PRO 6000 GPUs: Compass on the NorMuon update and NorMuon for seeds 250 to 252 at 262M tokens and seed 250 at 1.05B tokens, and Compass (N/N) and NorMuon at 262M tokens with three seeds at NorMuon's per-step decay and two at twice that decay, where the two methods differ by 0.0037 and 0.0061 nats. Pooled with the pairs of Table 3, Compass on the NorMuon update and NorMuon are within 0.0003 nats at 262M tokens (11 seeds, 95% CI $[-0.0026, 0.0033]$, lower on 5), and Compass on the NorMuon update is 0.0058 nats below NorMuon at 1.05B tokens (5 seeds, 95% CI $[-0.0247, 0.0130]$, lower on 2). The additional seeds therefore leave the conclusion of Table 3 unchanged: at these budgets, the two methods are close.

Larger evaluation sets. To reduce sampling noise, we re-evaluated the final models of the long-budget pairs on the first 16.8M tokens of the validation split, 128 times the usual 131,072 tokens, in non-overlapping windows of 1,024 tokens. On the seeds of Table 3, Compass on the NorMuon update is 0.0009 nats below NorMuon at 262M tokens (95% CI $[-0.0039, 0.0022]$) and 0.0073 nats below it at 1.05B tokens (95% CI $[-0.0309, 0.0163]$). On WikiText-2, the two methods differ by 0.004 and 0.011 nats at the two budgets. These differences are small compared with the spread of 0.17 to 0.18 nats across NorMuon's seeds, and adding the RTX PRO 6000 pairs leads to the same conclusions. The larger evaluation sets therefore agree with Table 3.

A decay rule chosen before training. We also trained the Muon- and NorMuon-based pairs at both budgets under a decay rule that we chose before training. The rule keeps the total decay $\sum_t \eta_t \omega$ fixed across budgets; per step, this is twice NorMuon's decay at 65.5M tokens, half at 262M tokens, and one eighth at 1.05B tokens. The rule raises the final loss of NorMuon and of Compass on the NorMuon update by 0.0072 and 0.0075 nats at 262M tokens (on all eight GH200 seeds) and by 0.039 and 0.050 nats at 1.05B tokens (four and three seeds), so Table 3 uses NorMuon's decay. Under the rule, Compass on the NorMuon update is 0.0004 nats below NorMuon at 262M tokens (nine seeds, 95% CI $[-0.0024, 0.0016]$), and the two differ by less than 0.0001 nats at 1.05B tokens (five seeds, 95% CI $[-0.0066, 0.0067]$).

At 262M tokens under the rule, the mean losses over nine seeds are 3.7088 for Compass (N/N), 3.7102 for Nesterov Muon, 3.7150 for Compass on the NorMuon update, and 3.7154 for NorMuon.

Compass (N/N) is below NorMuon on eight of the nine seeds (95% CI $[-0.0119, -0.0014]$, paired permutation test $p = 0.012$) and below Nesterov Muon on seven (95% CI $[-0.0059, 0.0031]$). At 1.05B tokens, Compass (N/N) and Nesterov Muon differ by 0.0050 and 0.0051 nats on the two seeds trained (252 and 253). In these comparisons, the pair with seed 250 ran on the RTX PRO 6000 GPUs, and we exclude one Compass run on the NorMuon update (262M tokens, seed 251) that lacks a matching NorMuon run. Under this rule, Compass (N/N) has the lowest mean loss at 262M tokens, and Compass on the NorMuon update stays close to NorMuon at both budgets.

A.3 Data

We use the `sample-10BT` subset of FineWeb-Edu. Each document goes to the training split (95%) or the validation split (5%), and we separate documents by the end-of-text token. The split for runs up to 65.5M tokens has 57M training and 3M validation tokens, and the larger splits of Section 4 are built in the same way from the same document stream. Training batches are random windows drawn with a seed-dependent generator, and the target at each position is the next token. Validation uses eight fixed batches drawn with seed 12345, with 32,768 tokens for the short runs and 131,072 for the long runs, all of the same shape, so their average equals the token-level mean loss.

The multilingual experiment uses French, German, Spanish, Italian, Portuguese, Polish, Dutch, and Indonesian from FineWeb2. We grouped documents by normalized URL, removed exact duplicates, and split the groups into training, validation, and test sets in proportions of 90%, 5%, and 5%. In the blocked order, a seed-specific permutation of the languages sets their order in the five cycles of eight 100-step blocks, and the shuffled order applies a second permutation to the same batches. We validate on 8,192 tokens per language every 100 steps. We fixed splits and batch orders before any training run.

A.4 Learning-rate selection and baselines

The selection rule in Section 4 applies to both widths, to 65.5M tokens, and to each data order in the multilingual experiment; ties go to the lower rate. The tuning seeds are 3500 and 3501 at width 1,024 and 3600 and 3601 at width 2,048, and the evaluation seeds are 250 to 257 and 260 to 267 at the two widths. At 65.5M tokens, we added eight evaluation seeds (258 to 265), and the 262M- and 1.05B-token pairs of Table 3 use seeds 267 to 274 and 251 to 254. The constant expert scale is the mean of $c\alpha$ over steps and expert matrices in a separate Compass run with the same settings. We computed it separately for each width, for the 65.5M-token setting, and for each data order; for multilingual training it is 0.940700 (shuffled) and 0.948149 (blocked).

Nesterov Muon and heavy-ball Muon share the shape factor $0.2\sqrt{\max(m, n)}$, the parameter groups, the AdamW learning rate, gradient clipping, schedule, and weight decay of Compass. For NorMuon, we ran the single-GPU reference implementation of Li et al. (2025) with its Newton-Schulz iteration in FP32; its own update scaling puts its learning rates in Table 7 on a different scale. Excitation follows the published utilization-based rule on heavy-ball Muon with exponent one, and its multiplier scales the full expert step, weight decay included. For Magma, we used a public AdamW-based implementation on the hidden matrices (masking probability 0.5, temperature 2, alignment averaging coefficient 0.9). AdamW uses the settings of Appendix C, both as a baseline and for the embedding, output-head, bias, and normalization parameters.

Table 7: Learning-rate sweeps. Each entry gives a learning rate and the mean final validation loss (nats) over the two tuning seeds; the last column gives the selected rate.

Setting	Method	Learning rate: mean tuning loss	Selected
8.2M, 1,024	Nesterov Muon	0.0005625: 5.58323, 0.001125: 5.46007, 0.00225: 5.61247	0.001125
8.2M, 2,048	Nesterov Muon	0.00034375: 5.43610, 0.0006875: 5.32134, 0.001375: 5.56518	0.0006875
8.2M, 1,024	Compass (N/H)	0.000625: 5.55381, 0.00125: 5.45141, 0.0025: 5.85329	0.00125
8.2M, 2,048	Compass (N/H)	0.000375: 5.41242, 0.00075: 5.31861, 0.0015: 5.66832	0.00075

Setting	Method	Learning rate: mean tuning loss	Selected
8.2M, 1,024	NorMuon	0.004: 5.58115, 0.008: 5.46068, 0.016: 5.64487	0.008
8.2M, 2,048	NorMuon	0.003125: 5.46190, 0.00625: 5.32107, 0.0125: 5.46869	0.00625
8.2M, 1,024	Excitation	0.0005625: 5.57691, 0.001125: 5.45991, 0.00225: 5.70183	0.001125
8.2M, 2,048	Excitation	0.00034375: 5.43100, 0.0006875: 5.31719, 0.001375: 5.62580	0.0006875
8.2M, 1,024	Magma	0.0006: 6.07928, 0.0012: 5.87716, 0.0024: 6.12660	0.0012
8.2M, 2,048	Magma	0.0004: 5.90850, 0.0008: 5.87356, 0.0016: 6.31308	0.0008
8.2M, 1,024	AdamW	0.0003: 6.02889, 0.0006: 6.10369, 0.0012: 6.59153	0.0003
8.2M, 2,048	AdamW	0.0002: 5.98742, 0.0004: 6.43759, 0.0008: 6.85597	0.0002
8.2M, 1,024	Heavy-ball Muon	0.0005625: 5.57384, 0.001125: 5.46451, 0.00225: 5.68255	0.001125
8.2M, 2,048	Heavy-ball Muon	0.00034375: 5.43027, 0.0006875: 5.32990, 0.001375: 5.53181	0.0006875
65.5M, 1,024	Compass (N/H)	0.000625: 4.48751, 0.00125: 4.34896, 0.0025: 4.34744	0.0025
65.5M, 1,024	Nesterov Muon	0.0005625: 4.52486, 0.001125: 4.35155, 0.00225: 4.31299	0.00225
8.2M, 1,024	Expert-only Nesterov	0.000625: 5.55031, 0.00125: 5.46093, 0.0025: 5.76266	0.00125
8.2M, 2,048	Expert-only Nesterov	0.000375: 5.40946, 0.00075: 5.31764, 0.0015: 5.77167	0.00075
8.2M, 1,024	Constant expert scale	0.000625: 5.54809, 0.00125: 5.46183, 0.0025: 5.77364	0.00125
8.2M, 2,048	Constant expert scale	0.000375: 5.41191, 0.00075: 5.32046, 0.0015: 5.60901	0.00075
8.2M, 1,024	Family factor only	0.000625: 5.55013, 0.00125: 5.46234, 0.0025: 5.78520	0.00125
8.2M, 2,048	Family factor only	0.000375: 5.40639, 0.00075: 5.31673, 0.0015: 5.67988	0.00075
8.2M, 1,024	Radius only	0.000625: 5.55394, 0.00125: 5.45810, 0.0025: 5.75885	0.00125
8.2M, 2,048	Radius only	0.000375: 5.41524, 0.00075: 5.31524, 0.0015: 5.69521	0.00075
65.5M, 1,024	Expert-only Nesterov	0.000625: 4.49086, 0.00125: 4.34898, 0.0025: 4.34528	0.0025
65.5M, 1,024	Constant expert scale	0.000625: 4.49121, 0.00125: 4.34978, 0.0025: 4.34448	0.0025
65.5M, 1,024	NorMuon	0.004: 4.52649, 0.008: 4.35955, 0.016: 4.30735	0.016
65.5M, 1,024	Compass (N/N)	0.000625: 4.49313, 0.00125: 4.33468, 0.0025: 4.32614	0.0025
8.2M, 1,024	Compass (N/N)	0.000625: 5.55654, 0.00125: 5.45029, 0.0025: 5.72406	0.00125
8.2M, 2,048	Compass (N/N)	0.000375: 5.41704, 0.00075: 5.31271, 0.0015: 5.60675	0.00075
65.5M, NorMuon decay	Compass (N/N)	0.00125: 4.31849, 0.00175: 4.29882, 0.0025: 4.30147	0.00175
65.5M, NorMuon decay	Compass (N/N), $2\times$ decay	0.00125: 4.30521, 0.00175: 4.28936	0.00175
65.5M, NorMuon decay	Compass (N/N), $4\times$ decay	0.00125: 4.29843	
65.5M, NorMuon decay	Nesterov Muon	0.00125: 4.31992, 0.00175: 4.30170, 0.0025: 4.30437	0.00175
65.5M, NorMuon decay	Nesterov Muon, $2\times$ decay	0.00125: 4.31241, 0.00175: 4.28181	0.00175

Setting	Method	Learning rate: mean tuning loss	Selected
65.5M, NorMuon decay	NorMuon, 2× decay	0.008: 4.33258, 0.016: 4.28653	0.016
65.5M, NorMuon decay	Compass on the NorMuon update	0.016: 4.30626, 0.022: 4.33721	0.016

For multilingual training, Compass, Nesterov Muon, heavy-ball Muon, expert-only Nesterov, and the constant expert scale are tuned for each order over $\{0.000625, 0.00125, 0.0025\}$ on seeds 3500 and 3501, and the evaluation runs use seeds 250 to 257. All but one of the ten selections are 0.00125. The exception is Nesterov Muon under blocked order, which picks 0.000625 by a margin of 7×10^{-6} nats in mean tuning loss (5.46943). NorMuon and Compass on the NorMuon update were tuned on the same seeds over $\{0.00283, 0.00566, 0.0113\}$, a grid on NorMuon's own scale. NorMuon chose 0.00283 (blocked) and 0.00566 (shuffled), and Compass on the NorMuon update chose 0.00566 for both orders. The NorMuon tuning runs used a second GH200 system (PyTorch 2.7, CUDA 12.8); all multilingual evaluation runs used PyTorch 2.12 and CUDA 13.2.

Figure 4 shows the validation loss during training at 65.5M tokens.

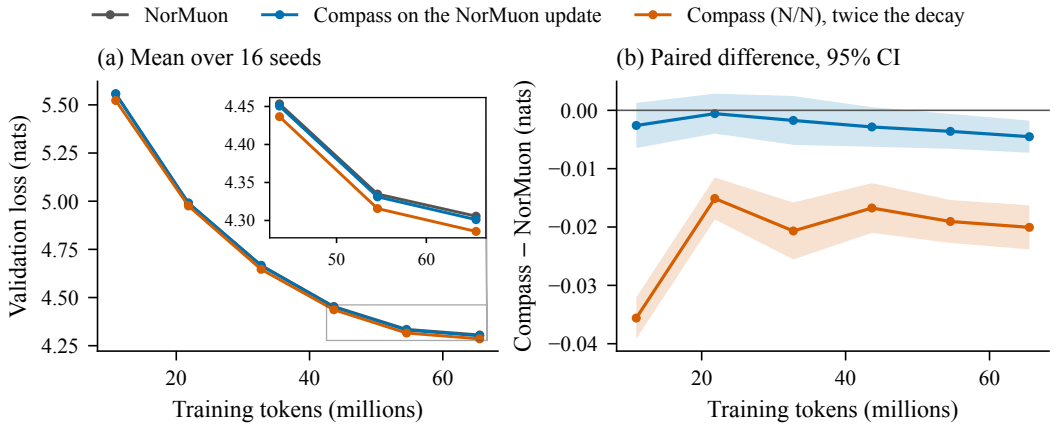


Figure 4: Validation loss during training at 65.5M tokens (width 1,024, context length 1,024; seeds 250 to 265, evaluated every 666 updates). (a) Mean over the 16 seeds; the inset enlarges the last three evaluations. (b) Compass minus NorMuon on the same seed, with pointwise 95% paired-t confidence bands. NorMuon and Compass on the NorMuon update use NorMuon's weight decay; Compass (N/N) uses twice NorMuon's per-step decay, as in Table 3.

B Paired comparisons and statistical tests

For each seed, the paired difference is the loss of Compass minus the loss of the baseline. We report means and sample standard deviations over seeds, and 95% confidence intervals and two-sided p-values from paired t-tests. For the main comparisons, we also give exact two-sided p-values from paired permutation (sign-flip) tests, which enumerate all 2^n sign assignments. Seeds are the independent replicates; results for different languages come from the same trained models and are therefore correlated.

Table 8 gives the paired t statistics and p-values of the ablations at 8.2M tokens, Table 9 lists every paired comparison, and Table 10 gives the WikiText-2 test loss of the 65.5M-token models. Table 11 adds the comparisons on all evaluation seeds at twice the decay, on the usual validation tokens and on the full validation split, and the paired WikiText-2 differences.

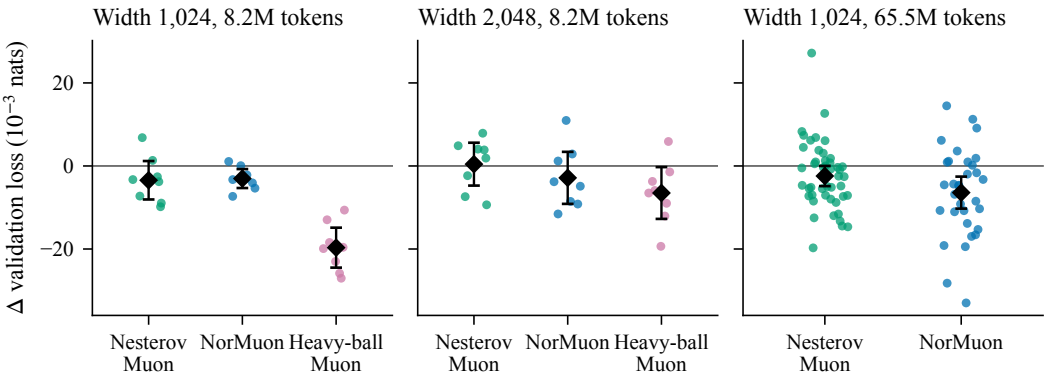


Figure 5: **Paired differences in final validation loss between Compass and each baseline.** Each point is one seed; diamonds and bars show the mean and 95% confidence interval. Negative values favor Compass. At 8.2M tokens, Compass uses Nesterov momentum on all matrices (N/N), with eight seeds per comparison. At 65.5M tokens, Compass (N/N), Nesterov Muon, and NorMuon all use twice NorMuon's per-step weight decay, with 46 seeds for Nesterov Muon and 32 for NorMuon (Section 4.3).

Table 8: Ablations at 8.2M tokens. Compass (N/H) minus each ablation in final validation loss (nats); negative values favor Compass. Intervals are 95% confidence intervals from paired t-tests over eight seeds, and t and p are the paired t statistic and its two-sided p-value.

Width	Ablation	Difference [95% CI]	t	p
1,024	Expert-only Nesterov	-0.00603 $[-0.01102, -0.00104]$	-2.86	0.024
2,048	Expert-only Nesterov	-0.00576 $[-0.01212, +0.00059]$	-2.14	0.069
1,024	Constant expert scale	-0.00749 $[-0.00934, -0.00564]$	-9.56	0.00003
2,048	Constant expert scale	-0.00298 $[-0.00872, +0.00276]$	-1.23	0.26

Table 9: Paired comparisons of final validation loss: Compass minus the comparison method (nats), 95% confidence interval, seeds favoring Compass, and two-sided paired t-test p-value. Settings give the token budget and the model width; $1\times$ and $2\times$ denote NorMuon's per-step weight decay and twice that value.

Setting	Compass	Comparison	Difference	95% CI	Wins	p
8.2M, 1,024	N/N	NorMuon	-0.00303	$[-0.00532, -0.00074]$	6/8	0.016
8.2M, 2,048	N/N	NorMuon	-0.00286	$[-0.00913, +0.00340]$	5/8	0.32
8.2M, 1,024	N/N	Nesterov Muon	-0.00344	$[-0.00807, +0.00119]$	6/8	0.12
8.2M, 2,048	N/N	Nesterov Muon	$+0.00044$	$[-0.00473, +0.00560]$	3/8	0.85
8.2M, 1,024	N/H	Nesterov Muon	$+0.00144$	$[-0.00415, +0.00704]$	5/8	0.56
8.2M, 2,048	N/H	Nesterov Muon	-0.00089	$[-0.00562, +0.00384]$	4/8	0.67
8.2M, 1,024	N/H	NorMuon	$+0.00186$	$[-0.00127, +0.00498]$	2/8	0.20
8.2M, 2,048	N/H	NorMuon	-0.00419	$[-0.00860, +0.00022]$	6/8	0.059
8.2M, 1,024	N/H	Excitation	-0.01033	$[-0.01425, -0.00641]$	8/8	4.3×10^{-4}
8.2M, 2,048	N/H	Excitation	-0.00582	$[-0.01004, -0.00159]$	8/8	0.014
8.2M, 1,024	N/H	Magma	-0.40414	$[-0.41412, -0.39417]$	8/8	3.6×10^{-12}
8.2M, 2,048	N/H	Magma	-0.50486	$[-0.53970, -0.47003]$	8/8	4.7×10^{-9}
8.2M, 1,024	N/H	AdamW	-0.58555	$[-0.59283, -0.57828]$	8/8	2.9×10^{-14}
8.2M, 2,048	N/H	AdamW	-0.61430	$[-0.63249, -0.59610]$	8/8	1.3×10^{-11}
8.2M, 1,024	N/H	Heavy-ball Muon	-0.01479	$[-0.01999, -0.00959]$	8/8	2.7×10^{-4}
8.2M, 2,048	N/H	Heavy-ball Muon	-0.00784	$[-0.01245, -0.00323]$	8/8	0.0051
8.2M, 1,024	N/H	Expert-only Nesterov	-0.00603	$[-0.01102, -0.00104]$	8/8	0.024
8.2M, 2,048	N/H	Expert-only Nesterov	-0.00576	$[-0.01212, +0.00059]$	7/8	0.069
8.2M, 1,024	N/H	Constant expert scale	-0.00749	$[-0.00934, -0.00564]$	8/8	2.9×10^{-5}
8.2M, 2,048	N/H	Constant expert scale	-0.00298	$[-0.00872, +0.00276]$	5/8	0.26
8.2M, 1,024	N/H	Family factor only	-0.00298	$[-0.00429, -0.00167]$	8/8	0.0010
8.2M, 2,048	N/H	Family factor only	-0.00151	$[-0.00566, +0.00264]$	5/8	0.42
8.2M, 1,024	N/H	Radius only	-0.00131	$[-0.00586, +0.00324]$	4/8	0.52
8.2M, 2,048	N/H	Radius only	$+0.00021$	$[-0.00405, +0.00448]$	3/8	0.91
65.5M, 1,024	N/H	Nesterov Muon	$+0.03141$	$[+0.02351, +0.03931]$	0/8	3.2×10^{-5}
65.5M, 1,024	N/H	NorMuon	$+0.04328$	$[+0.03669, +0.04988]$	0/8	1.1×10^{-6}
65.5M, 1,024	N/H	Expert-only Nesterov	-0.00132	$[-0.01038, +0.00775]$	4/8	0.74
65.5M, 1,024	N/H	Constant expert scale	$+0.00072$	$[-0.00796, +0.00940]$	4/8	0.85
65.5M, 1,024	N/N, $1\times$	NorMuon	-0.00587	$[-0.01024, -0.00151]$	13/16	0.012
65.5M, 1,024	N/N, $1\times$	Nesterov Muon	-0.01647	$[-0.02271, -0.01023]$	8/8	4.3×10^{-4}
65.5M, 1,024	N/N, $1\times$	Nesterov Muon, $1\times$	-0.00032	$[-0.00809, +0.00745]$	2/8	0.93
65.5M, 1,024	N/N, $2\times$	NorMuon	-0.02163	$[-0.02707, -0.01620]$	8/8	3.2×10^{-5}
65.5M, 1,024	N/N, $2\times$	Nesterov Muon, $1\times$	-0.01735	$[-0.02382, -0.01088]$	8/8	3.9×10^{-4}
65.5M, 1,024	N/N, $2\times$	Compass (N/N), $1\times$	-0.01703	$[-0.02032, -0.01374]$	8/8	5.6×10^{-6}
65.5M, 1,024	N/N, $2\times$	Nesterov Muon, $2\times$	-0.00269	$[-0.00839, +0.00301]$	6/8	0.30
65.5M, 1,024	N/N, $2\times$	NorMuon, $2\times$	-0.00790	$[-0.01747, +0.00167]$	5/8	0.092

Table 10: WikiText-2 test loss (nats) of the models trained for 65.5M tokens at context length 1,024, mean $\pm$ standard deviation over eight seeds. The evaluation scores all 286,178 test tokens and resets the context at document boundaries. The best mean is in bold.

Method	Momentum	WikiText-2 test loss
NorMuon	N/N	5.8705 ± 0.0337
Nesterov Muon	N/N	5.8785 ± 0.0265
Nesterov Muon, NorMuon decay	N/N	5.8438 ± 0.0220
Compass, NorMuon decay	N/N	5.8381 ± 0.0319
Compass, $2\times$ NorMuon decay	N/N	$\mathbf{5.8031} \pm 0.0350$
Compass on the NorMuon update	N/N	5.8592 ± 0.0395

Table 12 gives the multilingual replication. Pooled with the first set of runs, it gives 16 paired differences against Nesterov Muon at 0.000625 under blocked order, with mean -0.01995 nats

Table 11: Additional paired comparisons at 65.5M tokens. The validation rows use all evaluation seeds at twice NorMuon's per-step decay: seeds 250 to 281 for the comparisons with NorMuon and 250 to 295 for Nesterov Muon. The full-validation rows score the same models on the whole validation split of about 3M tokens, 23 times the usual 131,072, and their paired t-test p-values are below 0.001, 0.014, and 0.035; for the validation rows they are 0.002, 0.051, and 0.048. The WikiText-2 rows use the models of Table 10, and their exact two-sided paired permutation p-values are 0.023 and 0.031. Differences are Compass minus the comparison method, in nats, with 95% confidence intervals from paired t-tests; wins count the seeds on which Compass has the lower loss. "NorMuon update" marks Compass on the NorMuon update.

Loss	Compass	Comparison	Difference	95% CI	Wins
Validation	N/N, 2×	NorMuon, 2×	-0.00642	$[-0.01027, -0.00256]$	22/32
Validation	N/N, 2×	Nesterov Muon, 2×	-0.00242	$[-0.00485, 0.00001]$	30/46
Validation	NorMuon update, 2×	NorMuon, 2×	-0.00330	$[-0.00656, -0.00004]$	20/32
Full validation	N/N, 2×	NorMuon, 2×	-0.00744	$[-0.01075, -0.00412]$	25/32
Full validation	N/N, 2×	Nesterov Muon, 2×	-0.00280	$[-0.00499, -0.00060]$	29/46
Full validation	NorMuon update, 2×	NorMuon, 2×	-0.00303	$[-0.00583, -0.00023]$	20/32
WikiText-2	N/N, 2×	NorMuon	-0.0674	$[-0.1119, -0.0230]$	7/8
WikiText-2	N/N, 2×	Nesterov Muon, 1×	-0.0408	$[-0.0759, -0.0056]$	7/8

(95% CI $[-0.02992, -0.00998]$), lower on 14 of the 16 seeds (exact permutation $p = 0.0012$). The blocked-order gap therefore holds on new seeds.

Table 12: **Multilingual replication, seeds 1250 to 1257.** Held-out test loss of Compass minus each comparison method, in nats, with 95% confidence intervals from paired t-tests and exact two-sided paired permutation p-values. The four blocked-order comparisons were chosen before training, t is the paired t statistic, and the last row gives the shuffled order. The replication uses the data, schedule, and learning rates of Section 4.4 and differs from the first set of runs only in its seeds.

Order	Compass	Comparison	Difference	95% CI	Wins	t	p
Blocked	N/H	Nesterov Muon, 0.000625	-0.01719	$[-0.03641, +0.00204]$	7/8	-2.11	0.070
Blocked	N/H	Nesterov Muon, 0.00125	-0.00886	$[-0.02087, +0.00314]$	5/8	-1.75	0.133
Blocked	N/N	Nesterov Muon, 0.00125	-0.00449	$[-0.01879, +0.00981]$	6/8	-0.74	0.500
Blocked	N/H	Constant expert scale	-0.00223	$[-0.01341, +0.00894]$	5/8	-0.47	0.648
Shuffled	N/H	Nesterov Muon, 0.00125	$+0.00016$	$[-0.00250, +0.00283]$	2/8	0.14	0.922

NorMuon and Compass on the NorMuon update were trained after the other methods of Table 4, on the same data, seeds, and system, and Table 13 gives their comparisons. The lower block of Table 13 repeats these comparisons on the seeds of Table 12, with the learning rates fixed before training. Pooled with the first set of runs, it gives 16 paired differences against NorMuon under blocked order, with mean -0.02525 nats (95% CI $[-0.03654, -0.01396]$), lower on 13 of the 16 seeds (exact permutation $p = 0.0005$), and the mean difference favors Compass in all eight languages. The gap to NorMuon under blocked order therefore also holds on new seeds.

B.1 Language-specific results and training dynamics

Figure 6 shows the test loss difference for each language.

The fraction of experts that receive zero tokens is 0.47% under blocked order and 4.11% under shuffled order. The mean $c\alpha$ over the four training quarters is (0.899, 0.915, 0.957, 0.993) under shuffling and (0.910, 0.925, 0.963, 0.994) under blocking, and the family factors continue to vary across experts until the end of training.

Table 13: **Compass on the NorMuon update in the multilingual experiment.** Held-out test loss of Compass on the NorMuon update minus each comparison method, in nats, with 95% confidence intervals from paired t-tests and exact two-sided paired permutation p-values. The upper block uses seeds 250 to 257, and the lower block repeats the same comparisons on seeds 1250 to 1257 with the same learning rates. The first four comparisons were chosen before training, and t is the paired t statistic. The last row of each block gives the difference between the blocked and shuffled gaps relative to NorMuon.

Order	Comparison	Difference	95% CI	Wins	t	p
Seeds 250 to 257						
Blocked	NorMuon	-0.02346	$[-0.04128, -0.00563]$	6/8	-3.11	0.031
Blocked	Compass (N/H)	-0.01788	$[-0.03087, -0.00489]$	7/8	-3.25	0.016
Shuffled	NorMuon	$+0.00106$	$[-0.00079, +0.00292]$	3/8	1.35	0.219
Shuffled	Nesterov Muon	-0.00270	$[-0.00569, +0.00028]$	6/8	-2.14	0.086
Blocked minus shuffled	NorMuon	-0.02452	$[-0.04324, -0.00580]$	7/8	-3.10	0.023
Seeds 1250 to 1257						
Blocked	NorMuon	-0.02705	$[-0.04574, -0.00835]$	7/8	-3.42	0.016
Blocked	Compass (N/H)	-0.02439	$[-0.03985, -0.00894]$	8/8	-3.73	0.008
Shuffled	NorMuon	$+0.00016$	$[-0.00200, +0.00231]$	4/8	0.17	0.875
Shuffled	Nesterov Muon	-0.00286	$[-0.00505, -0.00067]$	7/8	-3.09	0.016
Blocked minus shuffled	NorMuon	-0.02721	$[-0.04521, -0.00920]$	7/8	-3.57	0.016

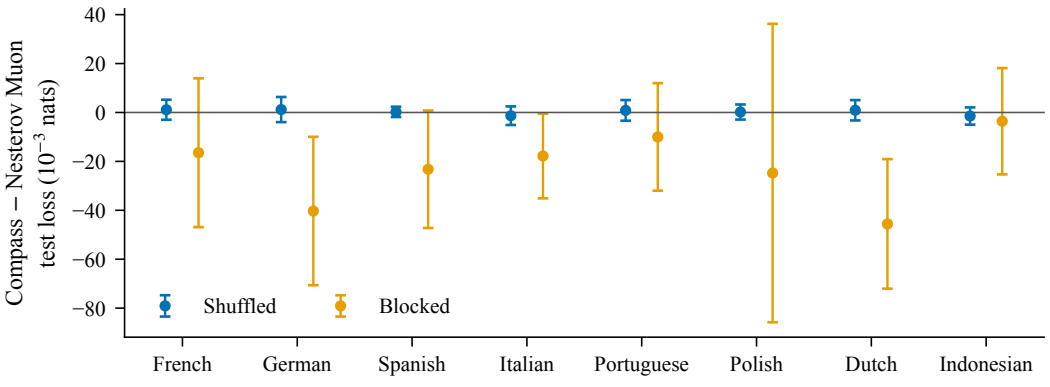


Figure 6: **Test loss by language.** Compass minus Nesterov Muon under shuffled and blocked order, with 95% confidence intervals over eight seeds. Results for different languages come from the same models. Under blocked order, all eight languages favor Compass, and under shuffled order, every mean difference is at most 0.0015 nats in magnitude.

C NUMERICAL DEFINITION OF COMPASS

Equations (1) to (7) and the schedule of Equation (9) define Compass in exact arithmetic. Here, we add the small constants of our code (Table 14), the order of operations (Section C.2), and properties of the two factors (Section C.5). All runs keep parameters, gradients, and optimizer states in FP32.

C.1 SYMBOLS AND CONSTANTS

Table 14: Symbols for one layer and one expert projection. A family contains the corresponding projection from every expert in that layer.

Symbol	Definition
t, T	Training step, starting at one, and total number of training steps.
e, E	Expert index and number of experts in the family.
r, m, n	Row index, number of rows and number of columns of an expert matrix.
$W_{e,t}$	Expert weight matrix before step t.
$G_{e,t}$	Expert gradient after global gradient clipping.
$M_{e,t}$	Momentum after its update at step t, with $M_{e,0} = 0$.
$D_{e,t}$	Input to the Newton-Schulz map, $G_{e,t} + \beta M_{e,t}$.
$\mathcal{N}_5$	Five iterations of the quintic map in Equation (11).
$s, O_{e,t}$	Shape factor $0.2\sqrt{\max(m,n)}$ and direction $\mathcal{N}_5(D_{e,t})$.
$z_{e,t}, a_{e,t}$	Cosine of the full matrix and its positive part plus ϵ_{score}.
$u_{e,t}, v_{e,t}, c_{e,t}$	Normalized expert score, capped score and family factor.
$z_{er,t}, b_{er,t}$	Row cosine and its positive part plus ϵ_{score}.
$q_{er,t}, h_{er,t}$	Capped row score and row factor after the schedule interpolation.
$\alpha_{e,t}$	Scalar radius computed from the weighted squared row norms.
$V_{e,t}$	Expert update direction, $c_{e,t}\alpha_{e,t}sO_{e,t}$.
$\eta_{\max}, \eta_t$	Peak matrix learning rate and its value at step t.
λ_t	Schedule multiplier $\eta_t/\eta_{\max}$.
β, ω	Muon momentum coefficient 0.95 and weight decay 0.01.
ϵ_{NS}	Normalization constant 10^{-7} of the Newton-Schulz map.
$\epsilon_{\cos}$	Constant 10^{-12} in the cosine denominator.
ϵ_{score}	Constant 10^{-6} added to the expert and row scores.
θ_t, g_t	Parameter updated by AdamW and its clipped gradient.
m_t, v_t	First and second AdamW moments.
$\widehat{m}_t, \widehat{v}_t, \tau_t$	Bias-corrected moments and the number of parameter updates.
$\beta_1, \beta_2, \epsilon_A$	AdamW constants 0.9, 0.95 and 10^{-8}.
η_t^{aux}, d_t	AdamW learning rate $\eta_{\max}^{\text{aux}}\lambda_t$ and denominator $\sqrt{\widehat{v}_t} + \epsilon_A$.

For matrices A and B of the same shape, $\langle A, B\rangle_F = \sum_{r,j} A_{rj}B_{rj}$ is the Frobenius inner product and $\|A\|_F = \sqrt{\langle A, A\rangle_F}$ its norm, and $[x]_+ = \max(x, 0)$ is the positive part. A family mean gives each expert matrix the same weight, and a row mean gives each row of a matrix the same weight.

C.2 ORDER OF OPERATIONS

We update the momentum buffer before forming the Nesterov input, so

$$D_{e,t} = G_{e,t} + \beta M_{e,t} = (1 + \beta)G_{e,t} + \beta^2 M_{e,t-1}. \tag{10}$$

The Newton-Schulz map divides its input by $\|D_{e,t}\|_F + \epsilon_{\text{NS}}$ and transposes it when $m > n$. It then runs five iterations with the coefficients of Jordan et al. (2024) and transposes the result back. Each iteration computes

$$A_k = X_k X_k^\top, \qquad X_{k+1} = 3.4445X_k + (-4.7750A_k + 2.0315(A_kA_k))\, X_k. \tag{11}$$

We compute both alignments from the unscaled direction $O_{e,t} = \mathcal{N}_5(D_{e,t})$ and the current clipped gradient, with $\epsilon_{\cos}$ added to each cosine denominator. For the full matrix,

$$z_{e,t} = \frac{\langle O_{e,t}, G_{e,t}\rangle_F}{\|O_{e,t}\|_F\|G_{e,t}\|_F + \epsilon_{\cos}}, \tag{12}$$

and the row cosine is the same expression for one row, with vector inner products and Euclidean norms. A row score is the positive part of the row cosine plus ϵ_{score}, divided by its mean over the rows and capped at one. Expert scores get the same offset and cap, and the two family normalizations divide by their means, clamped below at ϵ_{score}.

We compute inner products and norms in FP32. For family normalization, we convert values to double precision before computing the cosine and the two family means; the row computations stay in FP32. The row factors interpolate linearly between the capped row scores at the first step and one at the final step, where the learning rate is zero. Writing h_e for the row factors, the radius is

$$\alpha_e = \begin{cases} \sqrt{\dfrac{\sum_r h_{er}^2 \|O_{er}\|_2^2}{\max(\|O_e\|_F^2, 10^{-30})}}, & \|O_e\|_F^2 > 0, \\ 1, & \|O_e\|_F^2 = 0. \end{cases} \tag{13}$$

Weight decay is applied first, and the weight update is

$$W_e \mathrel{+}= -\big(\eta_t c_e\big)\, s\, \alpha_e O_e, \qquad s = 0.2\sqrt{\max(m, n)}. \tag{14}$$

The radius applies to the direction, and the family and shape factors enter the scalar coefficient. In the N/H configuration, attention and router matrices use heavy-ball momentum. Heavy-ball Muon also uses it for the expert matrices, with both Compass factors set to one.

Each layer has one expert family per projection; row scores are normalized within an expert matrix, and expert scores are normalized across the experts of a family.

C.3 AdamW update

Embeddings, the output head, biases, and normalization parameters use AdamW (Loshchilov & Hutter, 2019). For such a parameter θ_t with clipped gradient g_t, let τ_t count the steps up to t on which it had a gradient; the moments start at zero. On a step with a gradient, with $\beta_1 = 0.9$ and $\beta_2 = 0.95$,

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \tag{15}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t \odot g_t, \tag{16}$$

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^{\tau_t}}, \qquad \widehat{v}_t = \frac{v_t}{1 - \beta_2^{\tau_t}}, \tag{17}$$

where $\odot$ is elementwise multiplication. The step is $\widehat{m}_t / d_t$ with $d_t = \sqrt{\widehat{v}_t} + \epsilon_A$ and $\epsilon_A = 10^{-8}$, scaled by the learning rate $\eta_t^{\text{aux}} = \eta_{\max}^{\text{aux}} \lambda_t$, and the update uses decoupled weight decay 0.01. On a step without a gradient, the parameters and τ_t stay as they are. All Muon-based methods in a comparison use the same peak rate $\eta_{\max}^{\text{aux}}$ (Section 4 and Table 17), and the AdamW and Magma baselines use their tuned learning rate for all parameters.

C.4 Schedule

The schedule of Equation (9) starts at the peak learning rate with zero warmup and reaches zero at the final step. The row factors $h_{er,t} = 1 + \lambda_t(q_{er,t} - 1)$ therefore equal $q_{er,t}$ at the first step and one at the final step, while the family factor $c_{e,t}$ uses the current expert scores at every step. The momentum is also updated at the final step, where the parameter update and the decay are zero.

C.5 Properties of the scalar factors

The following properties hold in exact arithmetic with the constants defined above.

Family mean. For a family with $E\epsilon_{\text{score}} \le 1$, the family factors in Equation (3) satisfy

$$c_{e,t} > 0, \qquad \frac{1}{E} \sum_e c_{e,t} = 1. \tag{18}$$

Since $a_{e,t} \ge \epsilon_{\text{score}}$, the mean of a is at least ϵ_{score}, so $u_{e,t}$ has mean one and at least one entry is at least one. After capping, at least one entry has $v_{e,t} = 1$, so $E^{-1} \sum_e v_{e,t} \ge 1/E \ge \epsilon_{\text{score}}$. The final denominator is therefore the mean of v, which gives Equation (18). All families in our models satisfy the condition, since $E \le 64$. Individual factors can exceed one, and the total norm of the expert updates also depends on their matrix norms and radii.

Radius bounds. For $0 \le \lambda_t \le 1$, the row construction yields $0 < q_{er,t} \le 1$ and $1 - \lambda_t \le h_{er,t} \le 1$. Define

$$S_e = \sum_r \|O_{er,t}\|_2^2, \qquad H_e = \sum_r h_{er,t}^2 \|O_{er,t}\|_2^2. \tag{19}$$

Since $0 \le H_e \le S_e$, Equation (13) gives

$$0 \le \alpha_{e,t} \le 1. \tag{20}$$

Above the floor in the denominator of Equation (13), $(1 - \lambda_t)^2 S_e \le H_e \le S_e$ and hence $1 - \lambda_t \le \alpha_{e,t} \le 1$. The product $c_{e,t}\alpha_{e,t}$ can exceed one, since $c_{e,t}$ can exceed one.

Norm and direction. Above the floor,

$$\|\alpha_{e,t} O_{e,t}\|_F^2 = \alpha_{e,t}^2 S_e \tag{21}$$

$$= H_e \tag{22}$$

$$= \|\mathrm{diag}(h_{e,t}) O_{e,t}\|_F^2, \tag{23}$$

where $\mathrm{diag}(h_{e,t})$ is the diagonal matrix of row factors. The matrix $\alpha_{e,t} O_{e,t}$ therefore has the norm of the row-weighted matrix and the direction of $O_{e,t}$ (Figure 7).

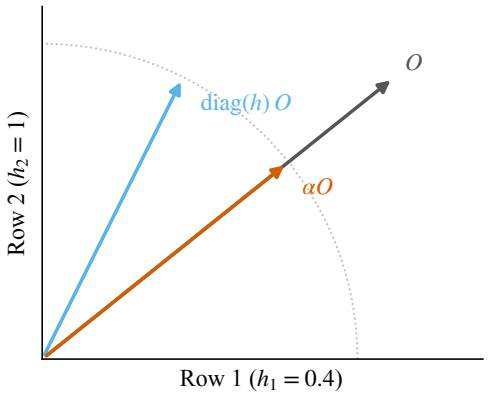


Figure 7: **The scalar radius for a matrix with two rows** and row factors $h = (0.4, 1)$. Row weighting gives $\mathrm{diag}(h)\, O$ and changes the direction. Compass applies αO, which has the norm of $\mathrm{diag}(h)\, O$ (dotted arc) and the direction of O. The family factor is omitted.

Let $O_{e,t} = L\Sigma R^\top$ be a singular value decomposition. For $\kappa = c_{e,t}\alpha_{e,t} > 0$,

$$V_{e,t} = sL(\kappa\Sigma)R^\top, \qquad \frac{V_{e,t}}{\|V_{e,t}\|_F} = \frac{O_{e,t}}{\|O_{e,t}\|_F}, \tag{24}$$

so the update keeps the singular vectors of $O_{e,t}$ and the ratios between its singular values. The weight change also includes weight decay.

Negative alignment. A matrix whose direction has a negative cosine with its gradient receives the score ϵ_{score}. When the other experts in its family have scores well above ϵ_{score}, its family factor and therefore its step are close to zero.

D COMPARISON WITH HEAVY-BALL MUON AT TWO WIDTHS

These runs train six-layer mixture-of-experts models from random initialization on FineWeb-Edu, with heavy-ball Muon as the baseline. They have their own learning-rate sweeps, so we report them separately from Section 4. The two sets of runs share seed numbers and selected learning rates. These runs used H100 GPUs, and their losses differ slightly from those of the same seeds in Table 1. Training windows come from the training part of a 60M-token sample of the `sample-10BT` subset, and both methods use the settings of Table 15.

Table 15: Settings of the runs at the two widths. Both methods use the same values.

Setting	Width 1,024	Width 2,048
Layers	6	6
Attention heads	16	16
Experts per layer	8	8
Selected experts per token	2	2
Expert hidden width	2,048	4,096
Sequences per step	16	16
Sequence length	256	256
Steps	2,000	2,000
Training tokens per run	8,192,000	8,192,000
Validation tokens per evaluation	32,768	32,768
Evaluation seeds	8	8

Both methods use auxiliary-loss-free load balancing (Wang et al., 2024). The router and its bias updates, the architecture, the FP32 arithmetic, the AdamW learning rate of 0.003, and the validation batches match Appendix A, and the schedule follows Equation (9) with $T = 2000$. The bias updates are independent of the Compass factors.

We selected the learning rates of both methods (Table 16) on tuning seeds before running the evaluation seeds.

D.1 METRICS AND TESTS

We use the paired statistics from Appendix B, with heavy-ball Muon as the baseline; a win is a seed on which Compass has the lower final validation loss. With eight seeds, the exact permutation test has $2^8 = 256$ sign assignments, so its smallest two-sided p-value is $2/256 \approx 0.0078$, the value at both widths in Table 16. Figure 8 shows each seed.

Table 16: **Comparison with heavy-ball Muon at two widths on FineWeb-Edu.** Final validation loss over eight evaluation seeds, with 8.192M training tokens per run. Values are mean $\pm$ sample standard deviation, and Δ is Compass minus heavy-ball Muon with its 95% confidence interval and statistic from a paired t-test. Heavy-ball Muon and Compass use learning rates 0.001125 and 0.00125 at width 1,024 and 0.0006875 and 0.00075 at width 2,048, selected on separate tuning seeds. The last column is the exact two-sided paired permutation p-value.

Width	Heavy-ball Muon	Compass	Δ	95% CI	t	Wins	p
1,024	5.4661 ± 0.0078	5.4551 ± 0.0075	-0.0110 ± 0.0065	$[-0.01638, -0.00555]$	-4.79	8/8	0.0078
2,048	5.3291 ± 0.0114	5.3192 ± 0.0094	-0.0099 ± 0.0049	$[-0.01393, -0.00578]$	-5.72	8/8	0.0078

Figure 9 shows the sweep over eight learning rates at width 2,048 on tuning seeds 2600 to 2604. For both methods, the best rate on these seeds is the one we used for the evaluation seeds. At these rates, Compass is lower on four of the five tuning seeds, by an average of 0.00754 nats.

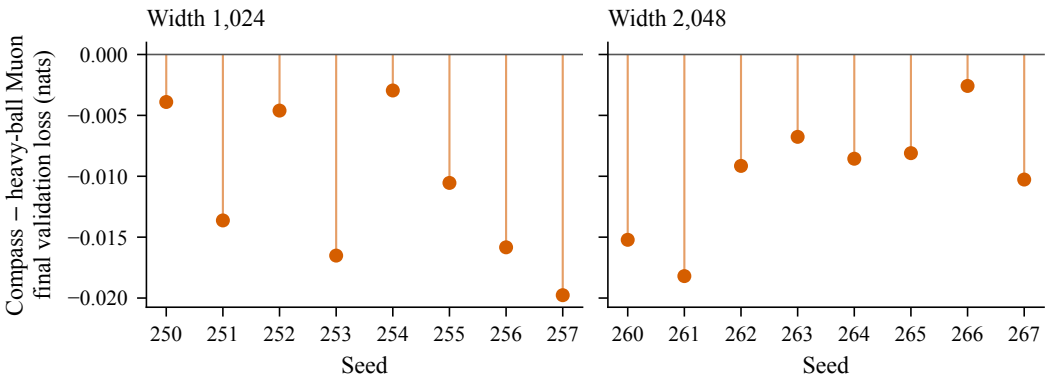


Figure 8: **Per-seed differences in the heavy-ball comparison.** Final validation loss of Compass minus heavy-ball Muon for each evaluation seed. All 16 seeds favor Compass.

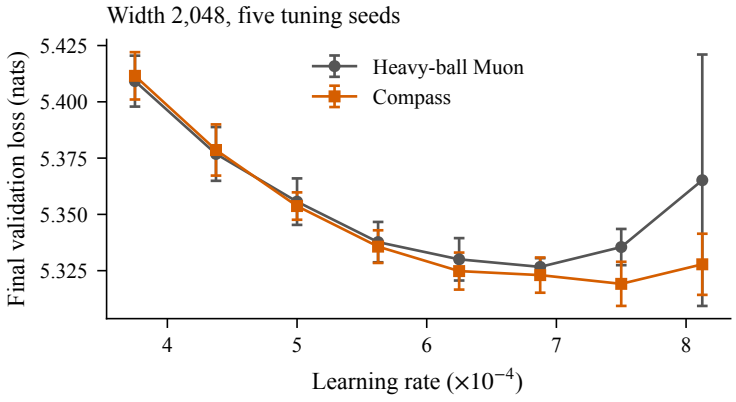


Figure 9: **Learning-rate sweep at width 2,048.** Points are means over five tuning seeds, and error bars are one sample standard deviation.

E CONTINUED PRETRAINING OF JETMOE

We apply Compass to JetMoE-8B, a released MoE checkpoint whose attention layers and feed-forward layers both contain experts (Shen et al., 2024). Three runs with different seeds lower its validation loss on FineWeb-Edu by about 0.043 nats after 8.2M training tokens (Table 18). These runs show that Compass applies to a released model of this size.

E.1 MODEL, DATA AND SETTINGS

JetMoE-8B has expert attention layers and expert feedforward layers, with about 8B total parameters and 2B active parameters per token (Shen et al., 2024). The runs start from the released JetMoE-8B checkpoint with a new optimizer state, and Compass updates both kinds of expert matrices. Packed expert tensors are split along the expert axis, so the last two axes form each expert matrix, and a family groups the matching projections within a layer and module. Router matrices use heavy-ball Muon, and embeddings, the output head, biases, and normalization parameters use AdamW.

The data are the `sample-10BT` subset of FineWeb-Edu, using the model's released tokenizer. Documents are split into training, validation, and test data in proportions 98%, 1%, and 1% before packing and are separated by an end-of-sequence token. The context length is 2,048, the training order is set by the seed, and the evaluation order uses seed 314159. Each validation evaluation scores 64 batches of eight sequences, for $64 \times 8 \times 2047 = 1{,}048{,}064$ target tokens, and the validation loss is the mean negative log-likelihood over these tokens. The best checkpoint of a run is the one with the lowest validation loss.

Table 17: Settings of the JetMoE runs. Learning rates are peak values.

Setting	Value
Parameters, gradients and optimizer states	FP32
GPUs per run	2×H100
Tokens per step	8,192
Sequence length	2,048
Accumulated micro-batches per GPU	2
Sequences per micro-batch per GPU	1
Steps	1,000
Tokens per run	8,192,000
Matrix learning rate	0.0001
AdamW learning rate	0.00002
Learning-rate schedule	Cosine, Equation (9), zero warmup
Gradient clipping	Global norm 1
Muon momentum	0.95
Weight decay	0.01
Validation interval (steps)	100
Seeds	991, 997, 1009
Software	Python 3.11, PyTorch 2.10.0, CUDA 12.8 Transformers 4.57.6, NumPy 1.26.4

Gradients are accumulated over micro-batches, averaged across the two GPUs in FP32 and then clipped, and automatic mixed precision is disabled. To save GPU memory, we update each expert family on one GPU, keep its momentum in host memory between steps, and broadcast the new parameters to the other GPU; the update is unchanged. Attention uses the deterministic implementation with TF32 off, activation checkpointing is on, and the update follows the order of operations in Appendix C.2.

E.2 PER-SEED RESULTS

All three seeds in Table 18 reach their lowest validation loss at step 100. An additional run with half the matrix learning rate (0.00005, seed 991) reaches a lower loss, shown in the last row of the table. A loss difference δ corresponds to a perplexity change of $100(1 - e^{\delta})$ percent.

Table 18: **Compass on JetMoE-8B.** Validation loss on FineWeb-Edu after 1,000 steps and 8.192M training tokens, with AdamW learning rate 0.00002. The first three rows use matrix learning rate 0.0001, and the last row is an additional run at 0.00005. The initial validation loss is 2.157508.

Seed	Matrix learning rate	Best loss	Final loss
991	0.0001	2.104669	2.115326
997	0.0001	2.102527	2.115145
1009	0.0001	2.104960	2.113939
Mean $\pm$ SD	0.0001	2.104052 ± 0.001329	2.114804 ± 0.000754
991	0.00005	2.072954	2.075305

Table 19: **Ablations of the alignment scores against heavy-ball Muon.** Final validation loss of each variant minus heavy-ball Muon at its selected learning rate; negative values favor the variant, and parentheses give wins over the number of seeds. Top: model widths and corpora; only settings with runs have entries. Bottom: variants and controls at width 1,024 on FineWeb-Edu with seeds 0 to 7, with 95% confidence intervals from paired t-tests. The scores a_e and b_{er} are the positive parts of the matrix and row cosines plus 10^{-6} (Section 3), and Cap marks capping at the family or row mean.

	FineWeb-Edu, width				DCLM
Variant	1,024	1,536	2,048	2,560	1,024
Family factor	-0.0057 (11/16)	-0.0041 (5/8)	-0.0049 (6/8)	-0.0070 (7/8)	-0.0041 (6/8)
Family, uncapped	-0.0039 (13/16)	-0.0039 (7/8)	-0.0064 (9/9)	$+0.0175$ (5/8)	
Family $\times$ row	-0.0059 (14/16)		-0.0078 (8/8)		-0.0075 (6/8)
Row weighting	-0.0034 (11/16)		-0.0010 (3/8)		-0.0069 (7/8)
Excitation	$+0.0035$ (2/8)		-0.0080 (7/8)		$+0.0029$ (2/8)

Level	Score	Cap	Difference [95% CI] (wins)
Family	a_e	yes	-0.00753 $[-0.01391, -0.00114]$ (6/8)
Family	a_e	no	-0.00393 $[-0.00896, +0.00109]$ (7/8)
Family	a_e, factors permuted	yes	-0.00066 $[-0.00457, +0.00326]$ (5/8)
Family	$\|G_e\|_F$	yes	-0.00336 $[-0.01035, +0.00363]$ (5/8)
Family	$[\cos(M_e, G_e)]_+$	no	-0.00419 $[-0.00866, +0.00028]$ (6/8)
Family	$1/a_e$	no	$+0.01812$ $[+0.00973, +0.02651]$ (1/8)
Family	a_e without mean-one normalization		$+0.19909$ $[+0.19317, +0.20500]$ (0/8)
Row	b_{er}	yes	-0.00482 $[-0.01016, +0.00052]$ (7/8)
Row	b_{er}	no	-0.00871 $[-0.01344, -0.00398]$ (8/8)
Row	$1/b_{er}$	no	$+0.45838$ $[+0.44986, +0.46691]$ (0/8)

F Ablations of the alignment scores

These runs vary how Compass alignment scores affect the update, with heavy-ball Muon as a baseline. They use a separate data split, H100 GPUs, and FP32 arithmetic, so absolute losses differ from those in Section 4; heavy-ball Muon reaches 5.352 at width 1,024 here.

Setup. The models have six layers, eight experts per layer, and top-2 routing, and they train with context length 256, 16 sequences per step, and 2,000 steps (8.192M training tokens). Widths 1,024, 1,536, 2,048, and 2,560 give 329.8M, 664.5M, 1,112.5M, and 1,673.8M total parameters. We train on a 60M-token FineWeb-Edu sample (95% training and 5% validation tokens) and on a DCLM-Baseline sample of the same size (Li et al., 2024). Parameters, gradients, and optimizer states are in FP32, and each run uses one NVIDIA H100 GPU with PyTorch 2.7. Heavy-ball Muon has the shape factor $0.2\sqrt{\max(m, n)}$ and the AdamW settings of the variants. We tuned its learning rate on a grid at each width and obtained 0.00125 at width 1,024 on both corpora and 0.000625 at the larger widths (the width-1,536 grid had the two rates 0.000625 and 0.00125). All variants use this rate. Variants that we also swept over two or three learning rates on seeds 0 to 2 picked the same rate. The heavy-ball grids include seeds that the comparisons also use. This favors the baseline.

Variants. The family-factor variant multiplies the orthogonalized update of expert e by c_e (Equation (3)) with the radius set to one, and its uncapped form uses u_e. Row weighting multiplies row r of O_e by $q_{er}/\overline{q}_e$ (Equation (4) with $\lambda_t = 1$). This changes the direction of the update, and the scalar radius of Compass keeps the direction. The combined variant (family $\times$ row) applies both. In these runs every matrix uses heavy-ball momentum, so O_e comes from M_e, and Excitation follows its published utilization-based equations. The controls replace the score with the gradient norm, the momentum cosine $\cos(M_e, G_e)$ or the reciprocal of a_e, permute the capped factors among the experts of each family at every step, or drop the mean-one normalization, which multiplies each expert step by a_e.

Results. Table 19 lists every difference. The combined variant lowers the loss in all three settings where we ran it, and its 95% confidence intervals lie below zero in each ($[-0.00883, -0.00302]$ at width 1,024, $[-0.01346, -0.00209]$ at width 2,048, and $[-0.01414, -0.00090]$ on DCLM). The family factor alone lowers the mean loss at all four widths and on DCLM, with intervals below zero at widths 1,024 and 2,560 ($[-0.00929, -0.00203]$ and $[-0.01058, -0.00341]$). The uncapped family factor lowers the loss up to width 2,048. At width 2,560, two of its seeds end 0.068 and 0.109 nats above heavy-ball Muon, and with the cap the same seeds change by $+0.0001$ and -0.0127. Row weighting alone lowers the loss at width 1,024 on both corpora. Excitation has a higher mean loss than heavy-ball Muon at width 1,024 on both corpora and comes close to the combined variant at width 2,048 (95% CI $[-0.01646, 0.00049]$).

The controls at width 1,024 use seeds 0 to 7 (bottom of Table 19). Permuting the capped factors among the experts of each family removes most of the benefit of the family factor, and the reciprocal score or dropping the mean-one normalization raises the loss sharply. The cosine factor is 0.00417 nats lower than a factor computed from gradient norms (95% CI $[-0.00757, -0.00076]$). In the uncapped form, the momentum cosine $\cos(M_e, G_e)$ and the update cosine $\cos(O_e, G_e)$ give nearly equal results (difference 0.00026 nats, 95% CI $[-0.00299, 0.00350]$). At the row level, both weightings lower the loss, and the reciprocal row score raises it.

Relation to Compass. Compass keeps the capped family factor and turns the capped row factors into a single scalar radius, preserving the orthogonalized direction that row weighting changes. Across these runs, alignment-based scaling lowers the loss against tuned heavy-ball Muon at both the expert and row levels, while the permuted and reciprocal controls remove the gain or raise the loss.

G THEORY

We state the setting, prove the perturbation bound of Theorem 1, convert gradient errors into score errors, study the Newton-Schulz map, and check the bounds numerically.

G.1 SETTING

We use the Frobenius inner product and norm of Appendix C for matrices and the Euclidean ones for rows. For a matrix X, X_r is its rth row. For a vector h, $\|h\|_\infty = \max_r |h_r|$, and $\mathrm{diag}(h)$ is the diagonal matrix with diagonal h. The proofs rely on the Cauchy-Schwarz inequality $|\langle X, Y\rangle| \le \|X\|\,\|Y\|$, the triangle inequality, and the reverse triangle inequality $\big|\|X\| - \|Y\|\big| \le \|X - Y\|$ (Horn & Johnson, 2013, Chapter 5).

We keep the constants $\epsilon_{\cos}$, ϵ_{score} and ϵ_{NS} of Table 14, and we write the floor 10^{-30} of Equation (13) as τ^2 with $\tau = 10^{-15}$. The results hold for any positive values of these constants with $E\epsilon_{\text{score}} \le 1$ and for any $\tau \ge 0$. The case $\tau = 0$ is the radius of Equation (5), without a floor. With the cosine $\cos_\epsilon(X, Y) = \langle X, Y\rangle/(\|X\|\,\|Y\| + \epsilon_{\cos})$ of Equation (12), the scores are

$$a_e = [\cos_\epsilon(O_e, G_e)]_+ + \epsilon_{\text{score}}, \qquad b_{er} = [\cos_\epsilon(O_{er}, G_{er})]_+ + \epsilon_{\text{score}}. \tag{25}$$

A bar denotes the mean over the family, or over the rows of one matrix for the row scores. With the capped score $\check{a}_e = \min(a_e, \overline{a})$, its mean $\mu = \overline{\check{a}}$, and the row means $\nu_e = \overline{b}_e$, the update is

$$c_e = \frac{\check{a}_e}{\mu}, \qquad q_{er} = \min\Big(1, \frac{b_{er}}{\nu_e}\Big), \qquad h_{er} = 1 + \lambda(q_{er} - 1), \qquad V_e = sc_e\Phi_{\tau,h_e}(O_e), \tag{26}$$

where $\Phi_{\tau,h}(0) = 0$ and

$$\Phi_{\tau,h}(O) = \frac{\|\,\mathrm{diag}(h)O\|_F}{\max(\|O\|_F, \tau)}\,O \qquad (O \ne 0). \tag{27}$$

This is the update of Section 3 and Appendix C.2. In Equation (3), $v_e = \min(1, a_e/\overline{a}) = \check{a}_e/\overline{a}$, so $\overline{v} = \mu/\overline{a}$ and

$$c_e = \frac{v_e}{\overline{v}} = \frac{\check{a}_e}{\mu}, \qquad \mu = \overline{a}\,\overline{v}. \tag{28}$$

The clamps of Appendix C.2 leave these values unchanged, because $\overline{a} \ge \epsilon_{\text{score}}$ and $\overline{v} \ge 1/E \ge \epsilon_{\text{score}}$ (Appendix C.5). For $O \ne 0$, Equation (13) gives

$$\alpha = \Big(\frac{\sum_r h_r^2\|O_r\|_2^2}{\max(\|O\|_F^2, \tau^2)}\Big)^{1/2} = \frac{\|\,\mathrm{diag}(h)O\|_F}{\max(\|O\|_F, \tau)}, \tag{29}$$

so $\alpha O = \Phi_{\tau,h}(O)$. For $O = 0$, the radius is one and $\alpha O = 0 = \Phi_{\tau,h}(0)$. Hence $V_e = sc_e\alpha_e O_e = sc_e\Phi_{\tau,h_e}(O_e)$, where $s > 0$.

The two updates in Theorem 1 use the same family of experts. An expert with a zero gradient belongs to the family. An expert without a gradient in a step does not, because Compass skips it and Compass on the NorMuon update gives it NorMuon's update without Compass factors (Appendix A). The matrices may be rectangular or rank deficient.

G.2 PROOF OF THE PERTURBATION BOUND

Lemma G.1 (Bounds on the family and row factors)**.** *For both families, $\overline{c} = 1$, $c_e > 0$, $0 < q_{er} \le 1$ and $0 < h_{er} \le 1$. Also $A, B \ge \epsilon_{\text{score}}$, and with δ_a and δ_b as in Theorem 1,*

$$|c_e - \widetilde{c}_e| \le \frac{\delta_a}{A}(1 + \widetilde{c}_e), \qquad \frac{1}{E}\sum_e |c_e - \widetilde{c}_e| \le \frac{2\delta_a}{A}, \tag{30}$$

$$|q_{er} - \widetilde{q}_{er}| \le \frac{2\delta_b}{B}, \qquad \|h_e - \widetilde{h}_e\|_\infty \le \frac{2\lambda\delta_b}{B}. \tag{31}$$

Proof. By the Cauchy-Schwarz inequality,

$$|\cos_\epsilon(X, Y)| \le \frac{\|X\|\,\|Y\|}{\|X\|\,\|Y\| + \epsilon_{\cos}} < 1, \qquad \text{so} \qquad \epsilon_{\text{score}} \le a_e,\, b_{er} < 1 + \epsilon_{\text{score}}. \tag{32}$$

Means and minima of numbers in this interval also lie in it. Hence $\overline{a}$, $\breve{a}_e$, μ and ν_e lie in this interval for both families, and $A, B \geq \epsilon_{\text{score}}$. Summing $c_e = \breve{a}_e/\mu$ over the family gives

$$\sum_e c_e = \frac{1}{\mu}\sum_e \breve{a}_e = \frac{E\mu}{\mu} = E, \tag{33}$$

so $\overline{c} = 1$, and $c_e > 0$ because $\breve{a}_e > 0$. Also $0 < q_{er} \leq 1$ because $b_{er}/\nu_e > 0$, and $h_{er} = (1-\lambda) + \lambda q_{er}$ lies between q_{er} and one because $\lambda \in [0,1]$.

A minimum changes by at most the change of its arguments. If $|x - x'| \leq \delta$ and $|y - y'| \leq \delta$, then $\min(x,y) \leq \min(x'+\delta, y'+\delta) = \min(x',y') + \delta$, and exchanging the two pairs gives

$$|\min(x,y) - \min(x',y')| \leq \delta. \tag{34}$$

The same argument proves this inequality for the maximum. Averaging $|a_j - \widetilde{a}_j| \leq \delta_a$ over the family, Equation (34) with $x = a_e$ and $y = \overline{a}$, and averaging again give, in turn,

$$|\overline{a} - \overline{\widetilde{a}}| \leq \delta_a, \qquad |\breve{a}_e - \widetilde{\breve{a}}_e| \leq \delta_a, \qquad |\mu - \widetilde{\mu}| \leq \delta_a. \tag{35}$$

Since

$$c_e - \widetilde{c}_e = \frac{\breve{a}_e - \widetilde{\breve{a}}_e}{\mu} + \widetilde{\breve{a}}_e\Big(\frac{1}{\mu} - \frac{1}{\widetilde{\mu}}\Big) = \frac{\breve{a}_e - \widetilde{\breve{a}}_e}{\mu} + \widetilde{c}_e\,\frac{\widetilde{\mu} - \mu}{\mu} \tag{36}$$

and $\mu \geq A$, we get

$$|c_e - \widetilde{c}_e| \leq \frac{\delta_a}{\mu} + \widetilde{c}_e\,\frac{\delta_a}{\mu} \leq \frac{\delta_a}{A}(1 + \widetilde{c}_e). \tag{37}$$

Averaging over e with $\overline{\widetilde{c}} = 1$ gives the second bound in Equation (30).

For the rows of one matrix, $q_{er} = y_{er}/\nu_e$ with $y_{er} = \min(b_{er}, \nu_e)$. As above,

$$|\nu_e - \widetilde{\nu}_e| \leq \delta_b, \qquad |y_{er} - \widetilde{y}_{er}| \leq \delta_b. \tag{38}$$

Equation (36), with y_{er}, ν_e and $\widetilde{q}_{er}$ in place of $\breve{a}_e$, μ and $\widetilde{c}_e$, gives

$$|q_{er} - \widetilde{q}_{er}| \leq \frac{\delta_b}{B}\,(1 + \widetilde{q}_{er}) \leq \frac{2\delta_b}{B}, \tag{39}$$

because $\nu_e \geq B$ and $\widetilde{q}_{er} \leq 1$. Finally, $h_{er} - \widetilde{h}_{er} = \lambda(q_{er} - \widetilde{q}_{er})$. □

Lemma G.2 (The radius map)**.** *For $\tau \geq 0$, $h, \widetilde{h} \in [0,1]^m$ and $X, Y \in \mathbb{R}^{m\times n}$,*

$$\|\Phi_{\tau,h}(X)\|_F \leq \|X\|_F, \tag{40}$$

$$\|\Phi_{\tau,h}(X) - \Phi_{\tau,\widetilde{h}}(Y)\|_F \leq 3\|X - Y\|_F + \|Y\|_F\|h - \widetilde{h}\|_\infty. \tag{41}$$

Proof. Let

$$u_h(X) = \|\operatorname{diag}(h)X\|_F, \qquad J(X) = \frac{X}{\max(\|X\|_F, \tau)} \quad (X \neq 0), \qquad J(0) = 0. \tag{42}$$

Then $\Phi_{\tau,h} = u_h J$ and $\|J(X)\|_F \leq 1$. For any vector k,

$$\|\operatorname{diag}(k)X\|_F^2 = \sum_r k_r^2\|X_r\|_2^2 \leq \|k\|_\infty^2\|X\|_F^2. \tag{43}$$

With $k = h$, this gives $u_h(X) \leq \|X\|_F$ and hence Equation (40). With $k = h - \widetilde{h}$ and the reverse triangle inequality, it gives $|u_h(Y) - u_{\widetilde{h}}(Y)| \leq \|h - \widetilde{h}\|_\infty\|Y\|_F$, so

$$\|\Phi_{\tau,h}(Y) - \Phi_{\tau,\widetilde{h}}(Y)\|_F = |u_h(Y) - u_{\widetilde{h}}(Y)|\,\|J(Y)\|_F \leq \|h - \widetilde{h}\|_\infty\|Y\|_F. \tag{44}$$

By the triangle inequality, it remains to show that $\|\Phi_{\tau,h}(X) - \Phi_{\tau,h}(Y)\|_F \leq 3\delta$ for $\delta = \|X - Y\|_F$. This difference is symmetric in X and Y, so we may assume $\rho = \|X\|_F \geq \rho' = \|Y\|_F$. If $Y = 0$, the bound follows from Equation (40), because $\|X\|_F = \delta$. Otherwise, $d = \max(\rho, \tau)$ and $d' = \max(\rho', \tau)$ are positive, and $d \geq d'$. Equation (34) for the maximum and the reverse triangle inequality give

$$0 \leq d - d' \leq \rho - \rho' \leq \delta. \tag{45}$$

From

$$J(X) - J(Y) = \frac{X - Y}{d} - \frac{d - d'}{d\,d'}\,Y \tag{46}$$

and $\|Y\|_F \le d'$, we get

$$\|J(X) - J(Y)\|_F \le \frac{\delta}{d} + \frac{\|Y\|_F}{d'} \cdot \frac{d - d'}{d} \le \frac{2\delta}{d}. \tag{47}$$

Finally,

$$\Phi_{\tau,h}(X) - \Phi_{\tau,h}(Y) = \big(u_h(X) - u_h(Y)\big)J(X) + u_h(Y)\big(J(X) - J(Y)\big), \tag{48}$$

where $|u_h(X) - u_h(Y)| \le u_h(X - Y) \le \delta$ and $u_h(Y) \le \rho' \le d$. The norm of this difference is therefore at most

$$\delta + d \cdot \frac{2\delta}{d} = 3\delta. \tag{49}$$

□

Proof of Theorem 1. Let $F_e = \Phi_{\tau,h_e}(O_e)$ and $\widetilde{F}_e = \Phi_{\tau,\widetilde{h}_e}(\widetilde{O}_e)$, so that $V_e = sc_eF_e$ and $\widetilde{V}_e = s\widetilde{c}_e\widetilde{F}_e$. Then

$$V_e - \widetilde{V}_e = sc_e(F_e - \widetilde{F}_e) + s(c_e - \widetilde{c}_e)\widetilde{F}_e, \tag{50}$$

and since $c_e > 0$,

$$\|V_e - \widetilde{V}_e\|_F \le sc_e\|F_e - \widetilde{F}_e\|_F + s|c_e - \widetilde{c}_e|\,\|\widetilde{F}_e\|_F. \tag{51}$$

Lemma G.1 gives $h_e, \widetilde{h}_e \in (0, 1]^m$. Lemma G.2 with $X = O_e$ and $Y = \widetilde{O}_e$, together with Equation (31), then gives

$$\|F_e - \widetilde{F}_e\|_F \le 3\delta_O + \frac{2\lambda R\delta_b}{B}, \qquad \|\widetilde{F}_e\|_F \le \|\widetilde{O}_e\|_F \le R. \tag{52}$$

We insert these bounds in Equation (51) and average over the family. With $\overline{c} = 1$ and Equation (30),

$$\frac{1}{E}\sum_e \|V_e - \widetilde{V}_e\|_F \le s\Big(3\delta_O + \frac{2\lambda R\delta_b}{B}\Big) + sR\,\frac{2\delta_a}{A}, \tag{53}$$

which is Equation (8). Lemma G.1 also gives $A, B > 0$. The proof compares two fixed sets of inputs, so the bound also holds when the scores depend on the same gradient as the direction. □

G.3 From gradient errors to score errors

Lemma G.3 (Errors of the cosine)**.** *For vectors or vectorized matrices x, y, x', y' and $\epsilon > 0$,*

$$\left|\frac{\langle x, y\rangle}{\|x\|\|y\| + \epsilon} - \frac{\langle x', y'\rangle}{\|x'\|\|y'\| + \epsilon}\right| \le \frac{2(\|y\|\|x - x'\| + \|x'\|\|y - y'\|)}{\|x\|\|y\| + \epsilon}. \tag{54}$$

Taking positive parts and adding ϵ_{score} to both cosines does not increase the left side.

Proof. Let $N = \langle x, y\rangle$, $P = \|x\|\|y\|$, $N' = \langle x', y'\rangle$, $P' = \|x'\|\|y'\|$ and $\Delta = \|y\|\|x - x'\| + \|x'\|\|y - y'\|$. The identities

$$N - N' = \langle x - x', y\rangle + \langle x', y - y'\rangle, \qquad P - P' = (\|x\| - \|x'\|)\|y\| + \|x'\|(\|y\| - \|y'\|) \tag{55}$$

and the Cauchy-Schwarz and reverse triangle inequalities give

$$|N - N'| \le \Delta, \qquad |P - P'| \le \Delta. \tag{56}$$

Since

$$\frac{N}{P + \epsilon} - \frac{N'}{P' + \epsilon} = \frac{N - N'}{P + \epsilon} + \frac{N'}{P' + \epsilon} \cdot \frac{P' - P}{P + \epsilon} \tag{57}$$

and $|N'| \le P'$, the left side of Equation (54) is at most

$$\frac{\Delta}{P + \epsilon} + \frac{\Delta}{P + \epsilon} = \frac{2\Delta}{P + \epsilon}. \tag{58}$$

The positive part $[t]_+ = \max(t, 0)$ satisfies

$$|[t]_+ - [t']_+| \le |t - t'| \tag{59}$$

by Equation (34) for the maximum, and adding the same constant to two numbers does not change their difference. □

Lemma G.3 with $x = O_e$, $y = G_e$, $x' = \widetilde{O}_e$, $y' = \widetilde{G}_e$ and $\epsilon = \epsilon_{\cos}$ gives

$$|a_e - \widetilde{a}_e| \le \frac{2\big(\|G_e\|_F\|O_e - \widetilde{O}_e\|_F + \|\widetilde{O}_e\|_F\|G_e - \widetilde{G}_e\|_F\big)}{\|O_e\|_F\|G_e\|_F + \epsilon_{\cos}}. \tag{60}$$

For the rows, the lemma and the bounds $\|O_{er} - \widetilde{O}_{er}\|_2 \le \|O_e - \widetilde{O}_e\|_F$ and $\|G_{er} - \widetilde{G}_{er}\|_2 \le \|G_e - \widetilde{G}_e\|_F$ give

$$|b_{er} - \widetilde{b}_{er}| \le \frac{2\big(\|G_{er}\|_2\|O_e - \widetilde{O}_e\|_F + \|\widetilde{O}_{er}\|_2\|G_e - \widetilde{G}_e\|_F\big)}{\|O_{er}\|_2\|G_{er}\|_2 + \epsilon_{\cos}}. \tag{61}$$

The maxima of these bounds over e and r bound δ_a and δ_b. For the direction, we fix the momentum $M_{e,t-1}$ of the previous step. Equation (10) gives $D_e - \widetilde{D}_e = (1+\beta)(G_e - \widetilde{G}_e)$, and Proposition G.1 below gives

$$\|O_e - \widetilde{O}_e\|_F \le \frac{2(4.25)^5(1+\beta)\|G_e - \widetilde{G}_e\|_F}{\min(\|D_e\|_F, \|\widetilde{D}_e\|_F) + \epsilon_{\mathrm{NS}}}, \tag{62}$$

which bounds δ_O. With these bounds, Theorem 1 bounds the error of the update by the errors of the clipped gradients, for any clipping rule. The bound includes the dependence of each factor on its own gradient.

The size of the bound depends on the norms and the scores. For example, $A = B = 0.2$, $R = 1$, $\delta_O = 0$, $\delta_a = \delta_b = 10^{-3}$ and $\lambda = 1$ give $s(0.01 + 0.01) = 0.02s$ in Equation (8). Replacing A and B by their lower bound $\epsilon_{\mathrm{score}}$ gives a valid but much larger bound.

G.4 The Newton-Schulz map

Proposition G.1 (Five Newton-Schulz iterations)**.** *Let $p(x) = 3.4445x - 4.7750x^3 + 2.0315x^5$ be the polynomial of Equation* (11)*, and let $P_5 = p \circ p \circ p \circ p \circ p$. Let $D \in \mathbb{R}^{m\times n}$ have the singular value decomposition $D = L\Sigma R^\top$ (Horn & Johnson, 2013, Chapter 2) with singular values σ_i, and let $\hat{\sigma}_i = \sigma_i/(\|D\|_F + \epsilon_{\mathrm{NS}})$. Then*

$$\mathcal{N}_5(D) = L\,\mathrm{diag}\big(P_5(\hat{\sigma}_i)\big)R^\top, \tag{63}$$

where $0 \le P_5(x) \le 1.21$ for $0 \le x \le 1.21$ and $P_5(x) > 0$ for $x > 0$. Hence $\mathcal{N}_5(D)$ has the rank of D, $\|\mathcal{N}_5(D)\|_{\mathrm{op}} \le 1.21$ and $\|\mathcal{N}_5(D)\|_F \le 1.21\sqrt{\mathrm{rank}(D)}$. For $D' \in \mathbb{R}^{m\times n}$,

$$\|\mathcal{N}_5(D) - \mathcal{N}_5(D')\|_F \le \frac{2(4.25)^5\|D - D'\|_F}{\min(\|D\|_F, \|D'\|_F) + \epsilon_{\mathrm{NS}}}. \tag{64}$$

Proof. Consider first $m \le n$, where no transpose is taken. Let $X = L\,\mathrm{diag}(x_i)R^\top$ with orthogonal L and R, where $\mathrm{diag}(x_i)$ is the $m \times n$ matrix with diagonal entries x_i. Then

$$XX^\top = L\,\mathrm{diag}(x_i^2)L^\top, \tag{65}$$

and one iteration of Equation (11) maps X to

$$F(X) = 3.4445X - 4.7750(XX^\top)X + 2.0315(XX^\top)^2X = L\,\mathrm{diag}\big(p(x_i)\big)R^\top. \tag{66}$$

The first iterate is

$$X_0 = \frac{D}{\|D\|_F + \epsilon_{\mathrm{NS}}} = L\,\mathrm{diag}(\hat{\sigma}_i)R^\top, \tag{67}$$

so five iterations give Equation (63). For $m > n$, the map runs on $D^\top = R\Sigma^\top L^\top$, and transposing the result gives the same formula. Repeated and zero singular values are allowed.

We write

$$p(x) = x\,k(x^2), \qquad k(y) = 2.0315y^2 - 4.7750y + 3.4445. \tag{68}$$

The discriminant of k is

$$4.7750^2 - 4(2.0315)(3.4445) = -5.189382 < 0, \tag{69}$$

so $k > 0$ and $p(x) > 0$ for $x > 0$. The derivative is

$$p'(x) = g(x^2), \qquad g(y) = 10.1575y^2 - 14.325y + 3.4445, \tag{70}$$

and the discriminant of g is

$$14.325^2 - 4(10.1575)(3.4445) = 65.25559 > 0. \tag{71}$$

The two roots $y_- < y_+$ of g are positive because their sum and product are positive. The leading coefficient of g is positive, so $g < 0$ exactly between the roots. On $[0, \infty)$, p therefore increases up to $\xi = \sqrt{y_-}$, decreases from ξ to $\sqrt{y_+}$, and increases after $\sqrt{y_+}$. Hence

$$p(x) \le \max\{p(\xi), p(1.21)\} \qquad (0 \le x \le 1.21). \tag{72}$$

Exact evaluation gives $p'(0.5545) > 0 > p'(0.5546)$, so $\xi \in (0.5545, 0.5546)$. The positive terms of p increase with x and the negative term decreases, so

$$p(\xi) \le 3.4445(0.5546) - 4.7750(0.5545)^3 + 2.0315(0.5546)^5 = 1.2028\ldots < 1.21. \tag{73}$$

Since also $p(1.21) = 0.9778\ldots < 1.21$, p maps $[0, 1.21]$ into itself, and so does P_5. Together with $p(x) > 0$ for $x > 0$, this proves the bounds on P_5. By Equation (63), the singular values of $\mathcal{N}_5(D)$ are the numbers $P_5(\hat{\sigma}_i)$. They lie in $[0, 1.21]$ because $0 \le \hat{\sigma}_i < 1$ (each $\sigma_i \le \|D\|_F$), and they are positive exactly when $\sigma_i > 0$. This gives the rank and the two norm bounds.

For Equation (64), we first bound p'. Since g is convex,

$$\max_{0 \le y \le 1.21^2} g(y) = \max\{g(0), g(1.21^2)\} = \max\{3.4445,\, p'(1.21)\} = 4.2448\ldots, \tag{74}$$

and

$$\min_{y \in \mathbb{R}} g(y) = 3.4445 - \frac{14.325^2}{4 \cdot 10.1575} = -1.6061\ldots. \tag{75}$$

Hence $|p'| \le 4.25$ on $[-1.21, 1.21]$, and the mean value theorem (Rudin, 1976, Chapter 5) gives

$$|p(x) - p(x')| \le 4.25\,|x - x'| \qquad (x, x' \in [-1.21, 1.21]). \tag{76}$$

Let $K_1 = Q_1 \Theta Q_1^\top$ and $K_2 = Q_2 \Gamma Q_2^\top$ be symmetric matrices with orthogonal Q_1, Q_2 and eigenvalues $\theta_i, \gamma_j \in [-1.21, 1.21]$ (Horn & Johnson, 2013, Chapter 2). Since $K_1^k = Q_1 \Theta^k Q_1^\top$ for every k,

$$p(K_1) = Q_1 p(\Theta) Q_1^\top, \qquad p(K_2) = Q_2 p(\Gamma) Q_2^\top. \tag{77}$$

With $C = Q_1^\top Q_2$, the matrices

$$Q_1^\top \big(p(K_1) - p(K_2)\big) Q_2 = p(\Theta)C - Cp(\Gamma), \qquad Q_1^\top (K_1 - K_2) Q_2 = \Theta C - C\Gamma \tag{78}$$

have the entries $(p(\theta_i) - p(\gamma_j))C_{ij}$ and $(\theta_i - \gamma_j)C_{ij}$. Orthogonal factors do not change the Frobenius norm, so Equation (76) gives

$$\|p(K_1) - p(K_2)\|_F \le 4.25\,\|K_1 - K_2\|_F. \tag{79}$$

For $X \in \mathbb{R}^{m \times n}$, the symmetric matrix

$$\mathcal{D}(X) = \begin{pmatrix} 0 & X \\ X^\top & 0 \end{pmatrix} \tag{80}$$

satisfies

$$\mathcal{D}(X)^3 = \mathcal{D}(XX^\top X), \qquad \mathcal{D}(X)^5 = \mathcal{D}\big((XX^\top)^2 X\big), \tag{81}$$

and $\mathcal{D}(X)^2$ is block diagonal with the blocks $XX^\top$ and $X^\top X$. Since $\mathcal{D}$ is linear,

$$p(\mathcal{D}(X)) = \mathcal{D}(F(X)), \qquad \|\mathcal{D}(X) - \mathcal{D}(Y)\|_F = \sqrt{2}\,\|X - Y\|_F. \tag{82}$$

An eigenvalue θ of $\mathcal{D}(X)$ satisfies $\theta^2 \le \|X\|_{\mathrm{op}}^2$, because θ^2 is an eigenvalue of $\mathcal{D}(X)^2$, whose eigenvalues are those of $XX^\top$ and $X^\top X$. If $\|X\|_{\mathrm{op}} \le 1.21$ and $\|Y\|_{\mathrm{op}} \le 1.21$, Equation (79) therefore gives

$$\|F(X) - F(Y)\|_F = \tfrac{1}{\sqrt{2}} \|p(\mathcal{D}(X)) - p(\mathcal{D}(Y))\|_F \le 4.25\,\|X - Y\|_F. \tag{83}$$

The kth iterates X_k of D and X'_k of D' have singular values in $[0, 1.21]$, by the first part of the proof. Five applications of Equation (83) give

$$\|\mathcal{N}_5(D) - \mathcal{N}_5(D')\|_F \le 4.25^5 \, \|X_0 - X'_0\|_F. \tag{84}$$

With $\rho = \|D\|_F$, $\rho' = \|D'\|_F$ and $\epsilon = \epsilon_{\mathrm{NS}}$,

$$X_0 - X'_0 = \frac{D - D'}{\rho + \epsilon} + \frac{(\rho' - \rho)\, D'}{(\rho + \epsilon)(\rho' + \epsilon)}, \tag{85}$$

and $\|D'\|_F = \rho'$ and $|\rho - \rho'| \le \|D - D'\|_F$ give

$$\|X_0 - X'_0\|_F \le \frac{\|D - D'\|_F}{\rho + \epsilon} + \frac{|\rho - \rho'|}{\rho + \epsilon} \le \frac{2\|D - D'\|_F}{\rho + \epsilon}. \tag{86}$$

Since $\rho \ge \min(\rho, \rho')$, this proves Equation (64). For $m > n$, the same argument applies to the transposes. □

The direction $\mathcal{N}_5(D)$ is in general not the polar factor of D (Higham, 2008, Chapter 8), whose nonzero singular values equal one. For example, a matrix of rank one with $\|D\|_F$ much larger than ϵ_{NS} has $\hat{\sigma}_1$ close to one, and $P_5(1) = 0.6964\ldots$. The results above use the actual values $P_5(\hat{\sigma}_i)$.

G.5 Numerical checks

We tested the inequalities of this appendix on several thousand random inputs, including zero directions, rank-deficient matrices, repeated singular values, and inputs near the numerical floors. Each tested inequality held within its numerical tolerance.