

AdamW and Muon reach similar SST-2 accuracy by different routes

Vatsalya Betala

Main finding. At the tested settings, hybrid Muon fit the training objective faster and had a flatter matched-loss neighborhood, but it did not improve final accuracy. AdamW gave better validation cross-entropy, calibration, and consistency across seeds.

Experimental design and measurements

I fine-tuned the 28.8M-parameter `prajjwal1/bert-small` on the full SST-2 training split and evaluated on its 872-example validation split. Each run used three epochs (6,315 updates), batch size 32, maximum length 128, 6% warmup followed by linear decay, and global gradient clipping at 1. Three paired seeds gave each AdamW–Muon pair the same classifier initialization and minibatch order. AdamW used learning rate 3×10^{-5} throughout. The Muon configuration used 10^{-3} on 24 internal attention/MLP matrices and auxiliary AdamW at 3×10^{-5} elsewhere; “Muon” below therefore means this hybrid. I set weight decay to zero because one coefficient combined with learning rates that differ by $33\times$ would create unequal decoupled shrinkage.

I used accuracy for the decision rule and cross-entropy (CE), Brier score, ECE, and confidence on mistakes for probability quality. Steps to fixed training-CE thresholds measure optimization speed without noisy wall-clock timing. Clipping frequency, relative update size, and gradient–update cosine describe the update actually applied; the latter two use the same 24 matrices for both optimizers. Validation CE minus CE on a fixed 4,096-example training subset is a simple overfitting diagnostic. Sharpness was measured separately because predictive performance does not determine local loss geometry.

Metric	AdamW	hybrid Muon
Final validation accuracy (%)	88.57 ± 0.18	88.69 ± 0.93
Final validation CE	0.418 ± 0.014	0.481 ± 0.020
ECE (10 equal-width bins)	0.0839 ± 0.0049	0.0931 ± 0.0080
Steps to rolling training CE ≤ 0.10	4743 ± 559	2958 ± 686
Train–validation CE gap	0.342 ± 0.015	0.449 ± 0.017
Fraction of updates gradient-clipped	0.990 ± 0.002	0.884 ± 0.002
Relative update on matched matrices	$(7.87 \pm 0.12) \times 10^{-5}$	$(37.73 \pm 0.06) \times 10^{-5}$
Gradient–descent-update cosine	0.293 ± 0.009	0.109 ± 0.002

Mean $\pm$ sample standard deviation over three paired seeds.

Which optimizer performed better?

There is no meaningful accuracy winner. Muon’s mean advantage was only 0.11 percentage points, smaller than its seed-to-seed variation, and paired exact McNemar tests were non-significant for every seed ($p = 0.23\text{--}0.50$). AdamW was better by final CE, calibration, confidence on errors, and consistency. Muon was better by training efficiency: it reached rolling CE 0.15 about 30% sooner and CE 0.10 about 38% sooner. Its updates on the matched matrices were $4.8\times$ larger and less aligned with the instantaneous gradient, evidence that orthogonalized momentum changed the trajectory. Its larger generalization gap suggests that faster fitting also produced more overconfidence or overfitting under the fixed three-epoch budget.

Which solution appears flatter?

I compared seed-1 checkpoints at matched training loss instead of only comparing the final steps. On 512 fixed examples, their base CE values were 0.07895 (AdamW) and 0.07861 (Muon). For ten paired random directions, perturbations were scaled to a fixed fraction of each tensor’s Frobenius norm; I recorded the worse of the $+\delta$ and $-\delta$ losses. At 1% radius, mean whole-model loss increase was 7.40×10^{-4} for AdamW versus 3.33×10^{-4} for Muon; at 10%, it was 0.0191 versus 0.00554. Restricting perturba-

tions to the 24 Muon-eligible matrices gave the same ordering. A five-step projected-gradient search on 256 examples also agreed, but its base losses were less closely matched and it quickly left the locally quadratic regime. Muon therefore *appears* flatter in this parameterization, not intrinsically: sharpness changes under parameter rescaling, and the smallest-radius ordering at the final, unmatched-loss checkpoints differed.

How reliable is this conclusion?

The safest conclusion is that the two configurations have different dynamics, not that one optimizer is generally superior. Performance used only three seeds, sharpness used one, and all experiments used one model, dataset, and untuned learning-rate pair on MPS, where exact determinism is not guaranteed. Clipping was active on most steps, so its shared threshold was itself influential.

In a larger study I would sweep learning rate, momentum/betas, weight decay, and clipping for each optimizer; compare equal-step and validation-matched checkpoints; repeat sharpness across seeds, subsets, and radii; and add scale-aware Hessian trace/top-eigenvalue estimates. I would then test several model sizes and tasks, including SFT and RLVR, and report trade-offs among task quality, steps, wall time, and optimizer memory rather than naming a winner from one operating point.