Riding the Polar Express to Convergence: Getting Attention on the Track to Stability

Benji Van Lienden, Nicolas Rault-Wang, Andrew Thein, Jason Trinh, Justin Yang
{bvanlienden, nraultwang, athein26, jasontrinh, justin_yang}@berkeley.edu

Abstract

Stable, low-cost optimization remains a central challenge in training Transformer language models. Muon orthogonalizes 2D momentum tensors to improve conditioning with Newton-Schulz, an iterative fixed-polynomial method; the Polar Express variant replaces Newton-Schulz with a GPU-friendly dynamic schedule of minimax-optimized polynomials, demonstrating faster and more stable convergence in low-precision training. We empirically study "cheap and stable" Muon configurations on GPT-2 Small: we tune Polar Express iteration/cushion settings and compare application scope (QKV, O, FFN vs V, O, FFN) against AdamW and Muon with Newton–Schulz. Muon-based optimizers consistently beat a tuned AdamW baseline and support wider stable learning-rate ranges. Baseline Polar Express settings give good conditioning; very aggressive cushions over-regularize despite low orthogonality error. Scope drives the larger differences: applying Muon to all matrices yields the broadest stability window, while restricting to value/output/FFN recovers most loss benefits at lower overhead but with a narrow stable LR range; within a fixed scope, the choice of Polar Express versus Newton–Schulz is secondary at this scale.

1 Introduction

Training modern Transformer-based language models at scale places strict requirements on the optimization stability and the numerical methods employed. Popular choices such as AdamW are robust and easy to tune, but they treat all parameters as flat vectors, ignoring that many weights in a Transformer are naturally matrix-shaped (e.g., attention and MLP projections). The Muon optimizer was proposed to leverage this structure: instead of updating a flattened tensor, Muon updates each weight matrix via a geometry-aware step that keeps the update well-conditioned by approximately orthogonalizing it.

Two recent developments motivate our focus on *efficient and stable* Muon configurations. First, Muon Polar Express (Muon+PE) [1] replaces the Newton–Schulz orthogonalization step with a GPU-friendly polynomial approximation, but introduces new hyperparameters (safety factor, iterations and cushion) that remain under-explored. Second, recent work on deployment scope [12] suggests that most of Muon's benefit comes from the value/output and feed-forward blocks (V, O, FFN). This raises the question of whether Muon should be applied to all matrices (QKV, O, FFN) or selectively. In principle, orthogonalizing fewer matrices reduces the per-step computational complexity, but it is unclear how this affects the stability of the training run.

In this work, we compare the standard Newton-Schulz implementation (Muon+NS) against the Polar Express variant (Muon+PE), examining both "Full" (all matrices) and "VO/FFN" scopes. We specifically investigate two questions:

(A) *Hyperparameter sensitivity*—how do loss and stability depend on Muon+PE iteration and cushion budgets, and can we find cheaper settings that maintain convergence?

(B) *Scope vs. stability*—given a good Muon+PE configuration, does restricting Muon to VO/FFN matrices impact stability and entropy collapse more than the choice of orthogonalization method?

Across these experiments, we find that Muon-based optimizers consistently outperform a well-tuned AdamW baseline in both loss and stability. Additionally, we show that most of this benefit can be recovered with moderate Muon+PE budgets and selective deployment. Muon applied only to VO/FFN blocks slightly improves the final loss of the full-scope variant, though it exhibits a much narrower stability window. Conversely, applying Muon to all matrices preserves a broad entropy-stable learning-rate range. We conclude that while scope dictates the stability-efficiency trade-off, the choice between Polar Express and Newton–Schulz is secondary in this regime.

2 Background and Related Work

2.1 Standard Optimization and Spectral Anisotropy

Training large Transformer language models typically uses adaptive gradient methods, primarily Adam and AdamW [8, 7]. These optimizers maintain per-element learning rates based on gradient statistics. While effective, this elementwise approach ignores correlations within parameter matrices.

Recent work by Wang et al. [12] shows that Adam-style optimizers create *spectral skew* in weight matrices: updates concentrate along a few dominant directions corresponding to frequent features or "head" patterns. This leaves most parameters—the "tail" of the spectrum—poorly conditioned and under-updated. In deep Transformers, this anisotropy can cause feature collapse where the model over-fits dominant modes while missing subtler patterns. This limitation has motivated *structured optimizers* that account for parameter space geometry.

2.2 Structured and Geometry-Aware Optimization

Several methods address elementwise optimizers' limitations by using information about the loss landscape's curvature or structure. **Shampoo** [4] is a key example. Unlike Adam, Shampoo treats neural network weights as tensors and preconditions gradients using Kronecker products of smaller matrices from marginal statistics. By approximating the inverse Hessian via these structured preconditioners, Shampoo captures correlations *within* layers.

While Shampoo focuses on preconditioning to approximate Newton steps, **Muon** [5] focuses on *spectral shaping*. Rather than estimating local curvature, Muon enforces orthogonalization. Given a gradient matrix G with SVD $G = U\Sigma V^\top$, Muon updates using $UV^\top$, constraining updates to the Stiefel manifold. This makes updates isotropic—every singular direction gets equal treatment, preventing Adam's spectral skew.

2.3 The Muon Family: Newton–Schulz and Polar Express

SVD is too expensive for large matrices. To avoid this, standard Muon approximates the polar factor $UV^\top$ using the **Newton–Schulz (NS)** iteration—a purely iterative matrix-multiplication method converging to the matrix sign function. We refer to this as Muon+NS throughout this paper; see Table 3 for additional optimizer notation.

Amsel et al. [1] introduce a refinement we call **Muon+PE** (Polar Express). Instead of Newton–Schulz, Muon+PE uses a GPU-friendly minimax polynomial approximation to the matrix sign operation. This gives deterministic compute costs and exposes a "cushion" hyperparameter controlling how aggressively small singular values are pushed into the target orthogonal band. While Amsel et al. focus on theoretically optimal polynomials, our Phase 1 experiments treat these coefficients as tunable hyperparameters to measure their effect on training stability.

Appendix A expands on the optimization theory underpinning Muon+NS and Muon+PE.

2.4 Architectural Focus: Associative Memories (VO/FFN)

A key distinction in our work is applying Muon differently to specific Transformer components. We focus on Value-Output (VO) and Feed-Forward Network (FFN) matrices. Mechanistic interpretability work suggests these layers function as **associative memories** [3], storing knowledge pairs while Query-Key (QK) mechanisms handle routing.

Wang et al. [12] provide empirical support for this split. They show that with heavy-tailed data distributions, Adam fails to learn "tail" patterns because gradients are dominated by frequent "head" features. Muon's orthogonalization forces equal updates across all directions, enabling learning of tail-end associations. Their NanoGPT ablations confirm that applying Muon only to VO and FFN parameters captures nearly all benefits of full Muon training. This finding motivates our comparison of global Muon against VO/FFN-targeted variants.

The architectural motivation for focusing on VO/FFN matrices stems from their role in the Transformer's residual stream. Elhage et al. [2] describe how attention heads write to the residual stream via the VO projection, while FFN layers perform nonlinear transformations that can be viewed as key-value lookups. Both operations involve large matrix multiplications where spectral properties directly affect information flow. When these matrices develop spectral skew under Adam, the model's ability to store and retrieve diverse patterns degrades. By enforcing spectral isotropy in VO/FFN while leaving QK matrices under standard optimization, we test whether the benefits of structured optimization are primarily localized to these memory-like components.

2.5 Efficiency and Distributed Training

While Muon avoids exact SVD, Newton–Schulz iteration still requires multiple global matrix multiplications. In distributed training with Tensor Parallelism, this needs frequent 'all-gather' operations to reconstruct full gradients, creating a communication bottleneck.

Khaled et al. [6] address this with **MuonBP** (Block-Periodic Muon), proposing local block-diagonal orthogonalization for most steps-requiring no communication, with periodic global steps to realign the spectrum. This solves the *hardware efficiency* problem of the update rule. Our work is complementary: we accept the fixed communication cost of standard Muon to focus on *algorithmic stability* (entropy collapse) and hyperparameter dynamics of the orthogonalization step itself.

2.6 Stability Diagnostics and Entropy Collapse

We ground our stability analysis in **attention-entropy collapse**, which serves as our primary stability diagnostic throughout this work. Zhai et al. [13] identify this as a primary failure mode: as learning rates increase, attention distributions become one-hot (minimizing entropy), freezing information routing. Rybakov et al. [11] emphasize tracking spectral norms and entropy as essential diagnostics. We use attention entropy as our primary stability signal to define "Maximum Stable Learning Rates" (MSLRs), investigating whether Muon+NS(Full) and Muon+PE(Full)'s spectral constraints naturally prevent this collapse compared to AdamW.

3 Methods

3.1 Training Pipeline

Following Amsel et al., we trained a randomly-initialized GPT-2 Small on 1 billion `fineweb` tokens [9]. We evaluated the five optimizers listed in Table 3, varying Polar Express hyperparameters (iterations T, cushion c, safety factor s_f) to enable the Phase 1 grid search (Sec. 4.1).

Each training run presented in Sec. 5 was executed in a single-GPU configuration. To control for GPU-specific effects we exclusively collected our final data with NVIDIA RTX A6000 GPUs. Appendix C provides additional hardware and configuration details for reproducibility.

To quantify uncertainty, each configuration was repeated with three distinct random seeds. Appendix D specifies detailed hyperparameters, software versions, full training recipe, and notes on reproducibility using our public GitHub repository.[1]

[1] https://github.com/nraultwang/cs182-project/tree/main

3.2 Optimizer Variants

Experimental runs compare the five optimizers defined in Table 3. Phase 1 (Sec. 4.1) finds optimal hyperparameters for the Polar Express subroutine characteristic of the Muon+PE(Full) and Muon+PE(VO/FFN) variants, while Phase 2 (Sec. 4.2) compares these five optimizers with respect to attention stability metrics, defined in Sec. 3.3. Appendix A explains the theoretical foundations for the Muon variants proposed by Amsel et al. and Keller et al.

3.3 Evaluation Framework

For each of the five optimizers, at each LR, we log the following metrics. These metrics are precisely defined in Appendix F, which also specifies our entropy-collapse threshold.

- **End-to-end metrics:** training and validation loss vs. tokens, perplexity, step time, tokens/sec.
- **Orthogonality and spectra:** `pe/ortho_err_before` and `pe/ortho_err_after`, per-layer orthogonality errors for sentinel attention layers (0, 5, 11), and SVD-based singular-value summaries for post-PE updates in those attention and VO/FFN blocks.
- **Attention health:** per-head attention entropy, entropy quantiles, fraction of queries with $\max_j A_{h,i,j} > 0.95$, and statistics of the pre-softmax logits $QK^\top/\sqrt{d_{\text{attn}}}$. We say a layer exhibits *attention-entropy collapse* when its final mean attention entropy satisfies the collapse criterion defined in Appendix F.
- **Scale and gradient flow:** norms of Q/K/V, attention output, and FFN weights and their gradients in layers 0, 5, and 11.

4 Experiments

We split the study into two phases. Phase 1 characterizes the practical hyperparameter sensitivity and cost–stability tradeoffs of the Muon+PE update on GPT–2 pretraining. Phase 2 asks how to deploy Muon+PE *selectively*—in particular on the value/output (V/O) and feedforward (FFN) sublocks—to obtain cheap and stable training. Detailed training protocols are provided in Appendix E.

4.1 Phase 1: Polar Express hyperparameter characterization

4.1.1 Hypotheses

- **H1 (Iteration budget saturation).** For Muon+PE(Full), small iteration budgets (e.g., 3 or 5 polynomial steps) and simple schedules (e.g., $[5]$, $[3]$, or short lists like $[5, 1]$ or $[3, 0]$) are sufficient to match the stability and validation loss of a high-budget reference configuration. Additional iterations beyond this regime yield diminishing returns while increasing wall-clock cost.
- **H2 (Cushion robustness).** Within a moderate range around the paper value, the spectral cushion (our `polar_cushion` hyperparameter) only weakly affects end-to-end loss, but has a measurable impact on orthogonality error and the singular-value spectrum of updates. We expect cushions near 0.020–0.030 to be safe, with more aggressive values compressing spectra more strongly but potentially increasing instability.

Design note on interaction. Iteration budget and cushion may interact in practice: increasing T tends to move updates closer to the polar target, while a larger cushion compresses the spectrum more aggressively. To avoid conflating these effects, our experimental design first selects representative iteration budgets and then sweeps cushion at fixed T.

4.1.2 Goal

Phase 1 selects a *conservative* (T, c) (e.g., $[5]$ and $c \approx 0.024$) that is clearly stable, which is then carried forward into Phase 2.

4.2 Phase 2: Cheap and stable Muon+PE(VO/FFN)

4.2.1 Hypotheses

- **H3 (Cheap Muon+PE(VO/FNN)).** A Muon+PE(VO/FFN) configuration achieves stability and validation loss close to Muon+PE(Full), while reducing wall-clock time (fewer matrices orthogonalized per step) and maintaining tokens/sec comparable to or better than Muon+NS.
- **H4 (Attention hardening under stress).** Under an aggressive learning rate near the stability boundary, both Muon+PE(Full) and Muon+PE(VO/FFN) maintain healthier attention behavior than AdamW and Muon+NS. Muon+PE(VO/FFN) should retain most of this "attention hardening" effect at lower cost.

4.2.2 Goal

Phase 2 evaluates whether we can obtain *cheap and stable* training by (i) restricting Muon to the value/output and feedforward sub-blocks (VO/FFN), and (ii) comparing this selective deployment to Muon(Full) and AdamW. We fix a single Polar Express configuration from Phase 1 (iteration schedule and cushion) and vary only the optimizer and scope of orthogonalization.

5 Results

5.1 Phase 1: Hyperparameter Characterization

In Phase 1 we run hyperparameter sweeps to test the effect of safety factor s_f, number of iterations T, and cushion c on optimizer behavior.

5.1.1 Safety Factor

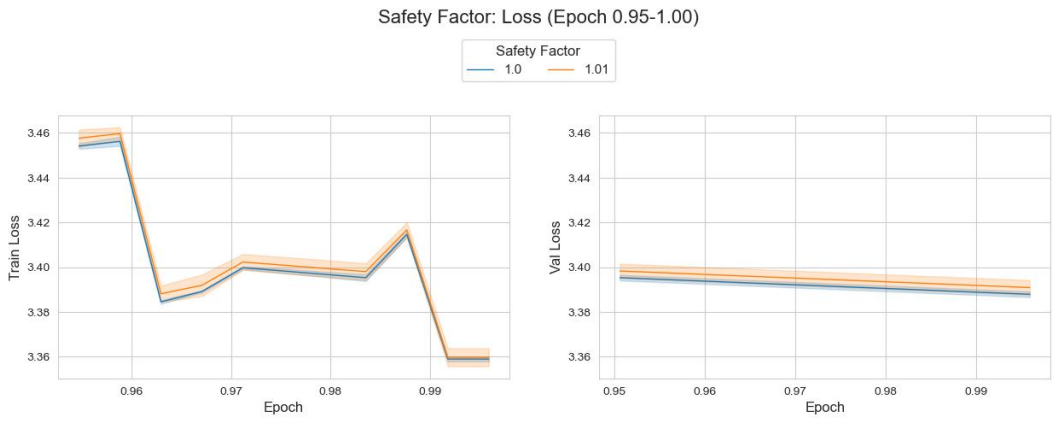

Figure 1: Training and validation loss for $s_f = 1.00$ and $s_f = 1.01$. The curves follow nearly identical paths, reinforcing that the safety factor has negligible effect on model behavior.

Across all monitored layers, the safety factor had no measurable effect on optimization behavior (Figure 1). Singular-value trajectories and both training and validation loss curves were nearly identical for $s_f = 1.00$ and $s_f = 1.01$, and error bars had considerable overlap throughout training.

5.1.2 Number of Iterations

The number of polar iterations T has a clear and consistent effect. Increasing T improves spectral stability and lowers orthogonality error, and reduces both train and validation loss (Figures 2 and 3). Empirically the ordering is $6 > 7 > 5 > 4 > 3$, with diminishing returns beyond $T = 6$. Alternating update schedules (e.g., switching T by layer or epoch) offered no benefit over a fixed T. The post-update singular-value densities become sharper and more coherent as T increases (Figure 2), aligning with the monotonic improvement in loss (Figure 3).

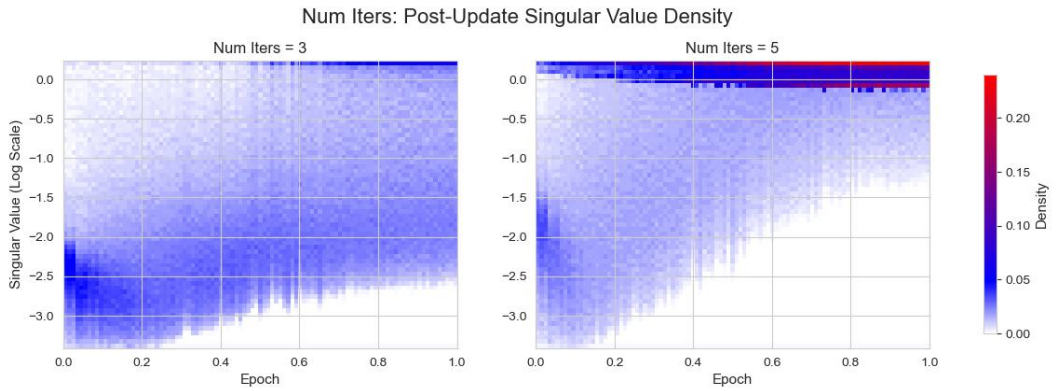

Figure 2: Post-update singular-value distributions for $T = 3$ and $T = 5$. Larger iteration counts yield sharper, more stable spectral structure and suppress low-magnitude drift.

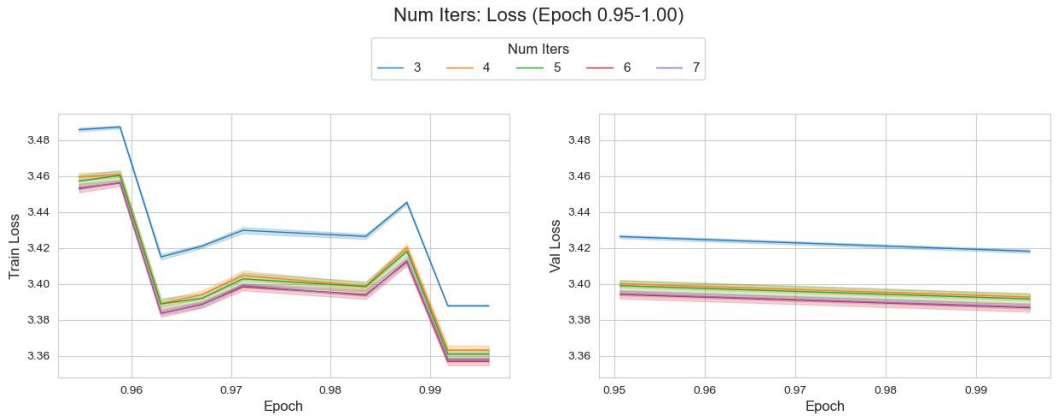

Figure 3: Training and validation loss for $T \in \{3, 4, 5, 6, 7\}$. Larger T consistently improves convergence, with $T = 6$ performing best and $T = 7$ producing no further gains

5.1.3 Cushion

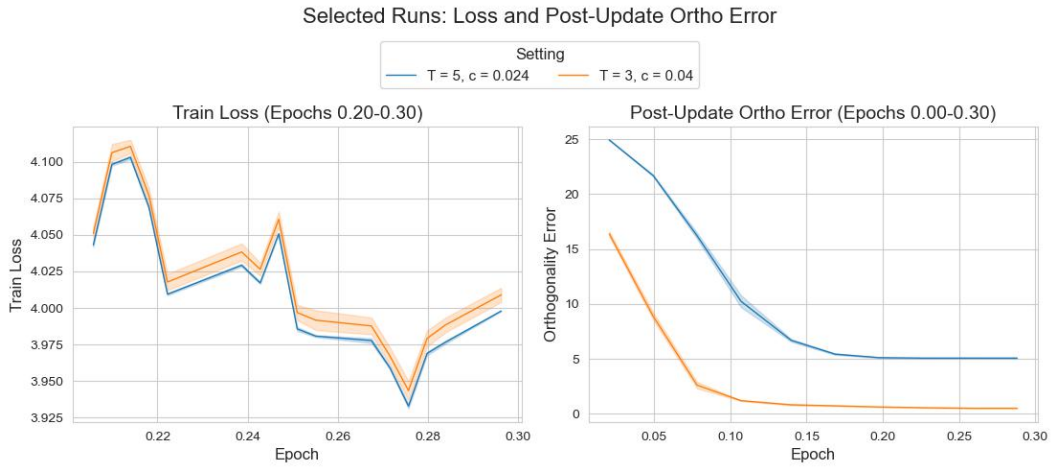

Figure 4: Train loss and post-update orthogonality error for $(T = 5, c = 0.024)$ versus $(T = 3, c = 0.04)$.

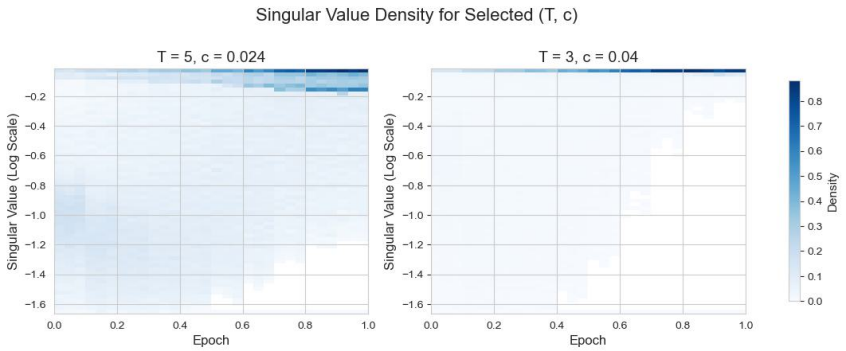

Figure 5: Singular value distributions for $(T = 5, c = 0.024)$ and $(T = 3, c = 0.04)$.

The effect of the cushion c is mixed. With a baseline $T = 5$, $c = 0.024$ we observe stable behavior, while a cheaper $T = 3$, $c = 0.04$ configuration achieves very similar loss curves (Figure 4). This suggests that a slightly larger cushion can compensate for fewer iterations, providing a lower-cost alternative to the $T = 5$, $c = 0.024$ baseline. However, despite a more isotropic post-update spectrum and lower orthogonality error, the cheaper configuration performs slightly worse than the baseline (Figure 5).

5.2 Phase 2: Attention-layer study

In Phase 2 we compare the five optimizers from Table 3 using the short 0.3-epoch learning-rate sweeps described in Section 4.2 and the attention-entropy metrics from Section 3.3. For each optimizer we choose a *baseline* LR by minimizing final train loss on its 0.3-epoch sweep and then examine attention entropy as LR is increased along the same sweep. From these curves we read off an entropy-based maximum stable learning rate (MSLR), defined as the largest LR for which none of the monitored layers (0, 5, 11) satisfy the collapse criterion in Appendix F. A summary of baseline LRs and MSLRs appears in Appendix I.

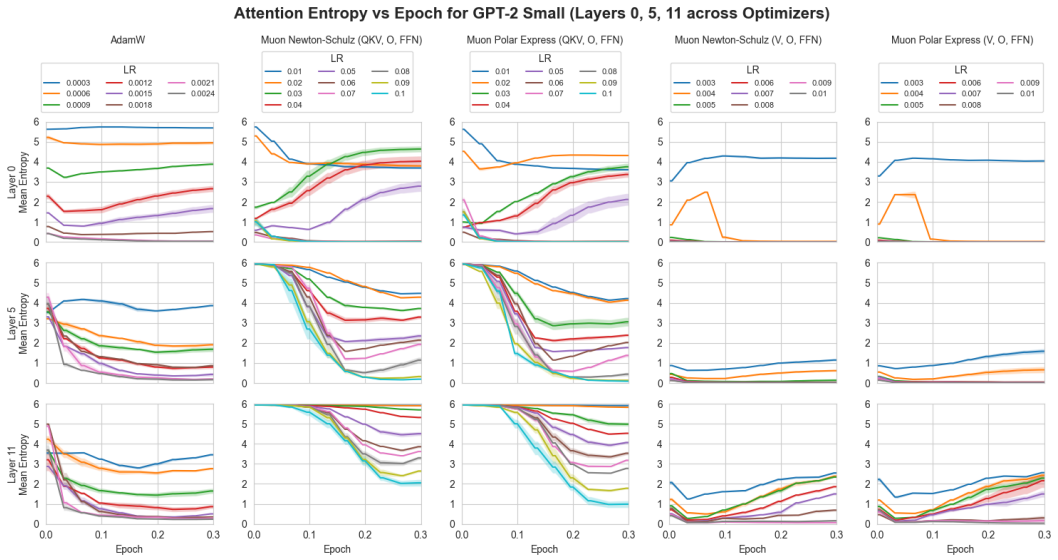

Figure 6: Attention entropy vs. epoch for GPT-2 Small in layers 0, 5, and 11 (rows) across five optimizer configurations (columns left-to-right): AdamW, Muon+NS(Full), Muon+PE(Full), Muon+NS(VO/FFN), and Muon+PE(VO/FFN). Each line shows the mean attention entropy for a particular learning rate, with shaded bands indicating the standard error across seeds.

5.2.1 Attention entropy and MSLR

Figure 6 shows distinct collapse patterns across optimizers under a common attention-entropy collapse criterion (final lower confidence bound ≤ 0.5 in any tracked layer). For AdamW, increasing the LR mainly destabilizes the deepest layer (layer 11), while the earliest layer stays comparatively high-entropy; AdamW remains entropy-stable up to roughly $4\times$ its baseline LR, giving an MSLR of about 0.0012 starting from a baseline of 0.0003. This depth ordering—with later layers collapsing before earlier ones under aggressive learning rates—matches the behavior reported for standard AdamW training by Zhai et al. [13].

For Muon(Full) variants, the pattern reverses: the earliest layer collapses first as LR increases, while deeper layers retain high entropy over a wide range. Both variants admit substantially larger entropy-stable LRs than AdamW, remaining stable up to about $5\times$ their baseline LR of 0.01 (MSLR ≈ 0.05), with little difference between Newton–Schulz and Polar Express in this regime. In other words, in our GPT-2 Small setting Muon inverts the depth ordering of collapse relative to AdamW: attention-entropy collapse consistently begins in layer 0, while layer 11 remains high-entropy throughout our LR sweeps.

Restricting Muon to VO/FFN yields a much narrower but sharply defined stability window. Starting from a baseline LR of 0.003, all three sentinel layers remain high-entropy only up to about $1\times$ this value; slightly larger LRs trigger a rapid attention-entropy collapse, beginning in the earliest layer. Thus both Muon(VO/FFN) variants have an MSLR very close to their baseline LR (MSLR ≈ 0.003), in contrast to AdamW and Muon(Full), whose entropy-stable region extends several multiples beyond their loss-optimal LR. At very aggressive LRs beyond the MSLR, some runs exhibit non-monotonic attention-entropy trajectories with an apparent "entropy rebound" but persistently high, noisy loss; we treat this non-learning regime and its implications in more detail in Appendix I.

5.2.2 Train and validation loss across optimizers

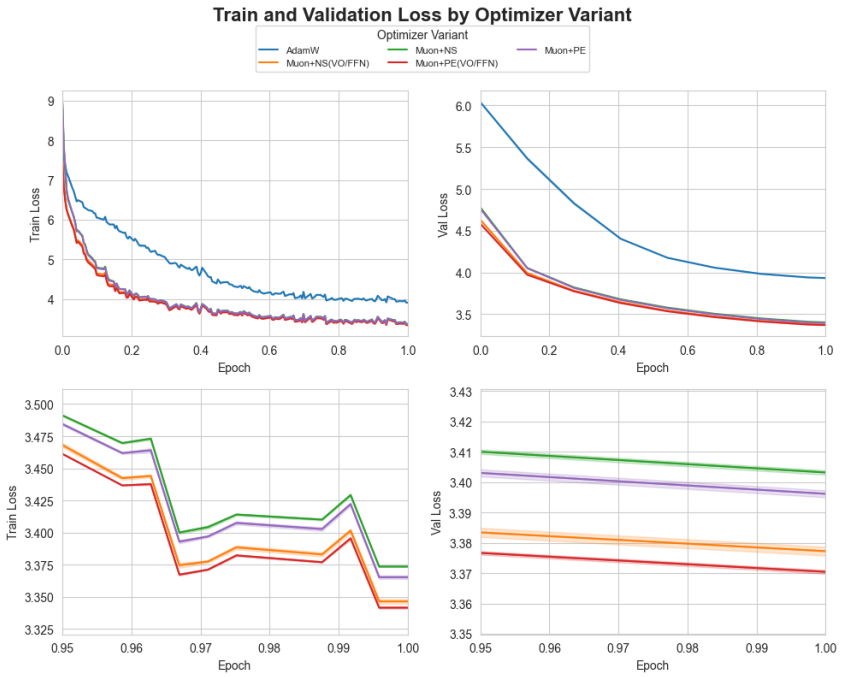

Figure 7: Train and validation loss for GPT-2 Small under different optimizer variants. Top: full 1-epoch runs; bottom: zoomed view of epochs 0.95–1.00 for the Muon variants only (AdamW omitted). All runs use optimizer-specific baseline learning rates selected from short 0.3-epoch LR sweeps (see Appendix I).

The loss curves in Figure 7 are consistent with prior work: all Muon-based optimizers substantially outperform AdamW in both training speed and final loss, so we focus on the smaller differences be-

tween Muon variants. The Muon configurations cluster tightly together, with only small differences in final loss. In the zoomed view, among the five baseline optimizers Muon+PE(VO/FFN) is slightly but consistently the best-performing variant at the end of training, with Muon+NS(VO/FFN) and the full-layer Muon variants close behind. Additional exploratory variants—including the "cheap" Muon+PE configuration and a split-QKV Muon+PE configuration—are reported in Appendix H and are not treated as core results.

Finally, our preliminary wall-clock comparisons were much noisier than the loss and entropy metrics: step times varied substantially across runs due to system-level effects. We therefore treat these timing differences as inconclusive and refer the reader to the Appendix J. for measurement details.

6 Discussion

6.1 Phase 1: Polar Express hyperparameters

Our Phase 1 sweeps indicate that the polar safety factor is effectively irrelevant in our regime. Runs with $s_f = 1.00$ and $s_f = 1.01$ have indistinguishable orthogonality error and loss curves, and we do not observe numerical precision issues in the matrix multiplications, so we treat s_f as a non-critical knob and fix it near the paper default.

For the cushion and iteration budget, we center our sweeps around the Polar Express paper's recommended cushion $c \approx 0.024$ (a numerically chosen value from their polynomial-design pipeline; see Appendix B for details) and vary both c and T. We find that a cheaper configuration with $T = 3$ iterations and $c = 0.04$ yields loss curves that are very close to a baseline with $T = 5$ and $c = 0.024$, despite using fewer Muon steps per update. Intuitively, the larger cushion makes the Polar Express step more aggressive in the sense that it shrinks the singular values more strongly toward a tighter band below 1, acting as a stronger spectral regularizer and partially compensating for the reduced iteration count. A heavier setting with $T = 6$, $c = 0.024$ gives the cleanest singular-value spectra and slightly better loss than $T = 5$, suggesting that, at this modest cushion, additional iterations still improve conditioning without erasing useful anisotropy in the updates. By contrast, very tight singular-value convergence (as in the $T = 3$, $c = 0.04$ setting) can over-regularize the update by flattening the spectrum and discarding anisotropic directions that are well aligned with the loss, so the best orthogonality metrics do not necessarily correspond to the best optimization performance. Increasing beyond $T = 6$ provides only marginal improvement in orthogonality and spectra. Taken together, these results largely confirm H1 and H2: a modest iteration budget ($T = 5$) and cushions near the Polar Express default yield good stability and loss, while more extreme cushions push the updates toward an over-regularized, overly isotropic regime without clear optimization gains.

6.2 Optimizer comparisons (Phase 2)

The Phase 2 results paint a consistent picture. First, any reasonable Muon configuration substantially outperforms AdamW in both optimization speed and final loss (Figure 7), and the Muon(Full) variants in particular admit significantly larger entropy-stable learning rates than AdamW (Table 4). In our GPT-2 Small / FineWeb setup, the AdamW baseline is entropy-stable only up to roughly $4\times$ its loss-optimal LR, whereas the Muon(Full) variants remain stable up to about $5\times$ their baseline LR and converge to much lower train and validation loss at their loss-tuned LRs, while the Muon(VO/FFN) variants trade away most of this extra LR headroom for cheaper deployment and a narrower stability window.

Second, the *scope* of Muon deployment (Full vs VO/FFN) is more consequential than the choice of orthogonalization method. Applying Muon to all attention and FFN matrices (QKV/O/FFN) yields a broad entropy-stable LR window and forgiving attention behavior, but requires orthogonalizing a large number of weight matrices per step. Restricting Muon to VO/FFN dramatically narrows the entropy-stable window—the MSLR essentially coincides with the baseline LR—yet the Muon(VO/FFN) variants still match or slightly improve on the Muon(Full) variants in final loss when run at their loss-tuned LRs (Figure 7). This suggests that, at least at this scale, most of the end-to-end benefit of Muon can be obtained by selectively hardening the value/output and FFN subblocks, while leaving Q and K under AdamW, consistent with recent ablation results showing that VO and FFN parameters are the primary contributors to Muon's advantage [12]. This partially confirms H3: Muon(VO/FFN) recovers most of the optimization benefit of Muon(Full) at lower cost,

but only within a narrow stability window that requires careful LR tuning. Additional ablations that more explicitly split the QKV block, including a split-QKV Muon+PE variant, are reported in Appendix H and treated as exploratory rather than core results.

A plausible contributor to the observed hypersensitivity of this hybrid configuration is our use of a single shared learning rate for both optimizer components. In the Muon(VO/FFN) runs, the shared LR of 0.003 lies between the AdamW baseline LR (0.0003) and the Muon baseline (0.01), so the Q and K blocks optimized with AdamW are effectively trained at a learning rate close to an order of magnitude above their loss-optimal setting, while the VO/FFN blocks optimized with Muon remain in a comfortable regime. This mismatch may partially explain why small increases in LR beyond 0.003 simultaneously trigger attention-entropy collapse and sharp loss degradation in the hybrid Muon(VO/FFN) variants. Disentangling the learning rates for the AdamW and Muon parameter groups is therefore a natural target for follow-up work.

Third, at fixed scope, the differences between Newton–Schulz and Polar Express are small. Across the Muon variants we study, Muon+NS and Muon+PE have nearly identical MSLRs and very similar loss curves, with Muon+PE showing only a slight, non-robust advantage over Muon+NS at the end of training. This effect is directionally consistent with the larger improvements reported for Polar Express in prior work [1], but much smaller in magnitude in our regime.

6.3 Depth-wise collapse patterns

Our attention-entropy analysis also reveals a qualitative difference in *where* instability first appears in depth. For AdamW, increasing the learning rate primarily destabilizes the deepest layer (layer 11), while the earliest layer remains comparatively high-entropy; this "top-down" collapse pattern is consistent with the behavior reported for standard AdamW training by Zhai et al. [13]. In contrast, for all of the Muon variants we study, attention-entropy collapse consistently begins in the earliest layer (layer 0), while the deepest layer (layer 11) remains high-entropy throughout our LR sweeps, even at learning rates where layer 0 has clearly collapsed.

A natural interpretation is that Muon reshapes the depth-wise sensitivity profile of the network. Under AdamW, deeper blocks operate closer to the logits and tend to see larger effective gradient scales, so they are the first to over-specialize and collapse as LR increases. Under Muon, the orthogonalization and norm-bounding of QKV/O/FFN updates appear to cap the sensitivity of deeper layers and shift the "weakest link" toward the input side: once layer 0 attention collapses, later layers can only recombine already low-entropy patterns rather than developing their own collapsed heads. For the Muon(VO/FFN) variants this effect is concentrated into a very narrow LR window, where small increases in LR beyond the MSLR simultaneously trigger early-layer collapse and a sharp increase in loss, whereas for all-matrices Muon the same mechanism operates over a much broader, more forgiving LR range. Overall, these observations only partially support H4: the Muon(Full) variants do harden attention relative to AdamW by maintaining high-entropy behavior over a wider LR range and exhibiting bottom-up rather than top-down collapse, but the Muon(VO/FFN) variants show lower attention entropy and collapse almost immediately once LR exceeds their baseline, so they do not provide the same degree of attention hardening despite their competitive loss.

7 Conclusion

We presented a systematic study of Muon-based optimizers for GPT-2 Small pretraining. Our findings indicate that: (1) Muon consistently outperforms AdamW final loss; (2) the *scope* of deployment dominates the choice of orthogonalization algorithm, where applying Muon to all matrices yields broad stability while restricting it to VO/FFN blocks yields higher accuracy; (3) Muon inverts the standard depth-wise collapse pattern, stabilizing deep layers while shifting instability to the input layer; and (4) cheaper Polar Express configurations ($T = 3$) can effectively match expensive baselines by aggressively cushioning the update.

For practitioners, this suggests that VO/FFN-only Muon is a strong high-efficiency default, while full-scope Muon remains safer for stability-critical regimes. We acknowledge limitations regarding our single model scale (GPT-2 Small), dataset (FineWeb-1B), and noisy timing infrastructure. Future work should extend these comparisons to larger models and longer training horizons to isolate the update rules that optimally balance conditioning with the preservation of useful anisotropy.

References

[1] Noah Amsel, David Persson, Christopher Musco, and Robert M. Gower. The polar express: Optimal matrix sign methods and their application to the muon algorithm, 2025. URL https://arxiv.org/abs/2505.16932.

[2] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

[3] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.emnlp-main.446/.

[4] Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pages 1842–1850. PMLR, 2018. URL https://proceedings.mlr.press/v80/gupta18a.html.

[5] 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/.

[6] Ahmed Khaled, Kaan Ozkara, Tao Yu, Mingyi Hong, and Youngsuk Park. Muonbp: Faster muon via block-periodic orthogonalization, 2025. URL https://arxiv.org/abs/2510.16981.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

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

[9] 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.

[10] Robert Zemeckis (Director). The polar express. Motion picture, 2004. Burbank, CA: Warner Bros. Pictures.

[11] Oleg Rybakov, Mike Chrzanowski, Peter Dykas, Jinze Xue, and Ben Lanir. Methods of improving llm training stability, 2024. URL https://arxiv.org/abs/2410.16682.

[12] Shuche Wang, Fengzhuo Zhang, Jiaxiang Li, Cunxiao Du, Chao Du, Tianyu Pang, Zhuoran Yang, Mingyi Hong, and Vincent Y. F. Tan. Muon outperforms adam in tail-end associative memory learning, 2025. URL https://arxiv.org/abs/2509.26030.

[13] Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Josh Susskind. Stabilizing transformer training by preventing attention entropy collapse, 2023. URL https://arxiv.org/abs/2303.06296.

A Theoretical foundations of Muon and Muon+PE

We briefly summarize the matrix-analytic foundations behind Muon and Muon+PE in terms of the singular value decomposition (SVD), which makes it clear that these methods act on the singular-value spectrum of updates.

Let $X \in \mathbb{R}^{m\times n}$ have compact SVD

$$X = U\Sigma V^\top, \tag{1}$$

where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ have orthonormal columns, $\Sigma \in \mathbb{R}^{r\times r}$ is diagonal with entries $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$, and $r = \mathrm{rank}(X)$. Many matrix functions applied to X or to $X^\top X$ reduce to applying a scalar function to the singular values; for example,

$$X^\top X = V\Sigma^2 V^\top, \qquad f(X^\top X) = V f(\Sigma^2) V^\top \tag{2}$$

for any scalar function f that is well-defined on the spectrum.

From this SVD viewpoint, "orthogonalizing" an update matrix G_t amounts to replacing its singular values with numbers closer to 1 while preserving (or only mildly perturbing) its singular directions. If $G_t = U_t\Sigma_t V_t^\top$, then an idealized orthogonalized update would have the form

$$\tilde{G}_t = U_t\,\tilde{\Sigma}_t\,V_t^\top, \tag{3}$$

where the diagonal entries of $\tilde{\Sigma}_t$ satisfy $\tilde{\sigma}_i \approx 1$ for all i. Muon+NS and Muon+PE approximate this effect without computing SVDs explicitly, by applying iterative or polynomial transformations to G_t that implicitly act on the singular values.

A.1 Muon+NS: Newton–Schulz polar iteration

In Muon+NS we seek the polar factor of a rectangular update $X \in \mathbb{R}^{m\times n}$,

$$Q \;=\; X\,(X^\top X)^{-1/2}, \qquad Q^\top Q = I,$$

without forming SVDs. A classical approach is the single–sequence Newton–Schulz (NS) iteration for the polar decomposition,

$$X_{k+1} = \frac{1}{2}\,X_k\big(3I - X_k^\top X_k\big), \qquad X_0 = X, \tag{4}$$

optionally after scaling X_0 so that $\|I - X_0^\top X_0\|_2 < 1$ to ensure quadratic convergence.

Writing $X_k = U\Sigma_k V^\top$ and treating U, V as approximately fixed during the iteration,

$$X_k^\top X_k \;=\; V\,\Sigma_k^2\,V^\top \quad\Longrightarrow\quad X_{k+1} \;=\; U\left[\frac{1}{2}\,\Sigma_k\big(3I - \Sigma_k^2\big)\right]V^\top,$$

so each singular value obeys the scalar update

$$\sigma \;\mapsto\; \frac{1}{2}\,\sigma\,(3-\sigma^2) \;=\; \frac{3}{2}\sigma \;-\; \frac{1}{2}\sigma^3. \tag{5}$$

The coefficients $\frac{3}{2}$ and $-\frac{1}{2}$ in (5) come directly from expanding the prefactor $\frac{1}{2}$ in the matrix iteration (4):

$$\frac{1}{2}\,\sigma(3-\sigma^2) = \left(\frac{1}{2}\cdot 3\right)\sigma - \left(\frac{1}{2}\cdot 1\right)\sigma^3 = \frac{3}{2}\sigma - \frac{1}{2}\sigma^3.$$

The map (5) contracts σ toward 1 (after the usual preprocessing that keeps the spectrum in a safe interval), thereby flattening the singular-value profile while approximately preserving the singular directions U and V. This clarifies NS as a spectrum-shaping procedure that moves X toward the Stiefel manifold $\{Y : Y^\top Y = I\}$ using only matrix multiplications.

A.2 Muon+PE: fixed polynomial spectrum shaping

Muon+PE replaces the cubic NS map inside Muon with a higher-order polynomial that is designed to approximate the matrix sign / polar operation on a safeguarded spectral interval $[\ell, 1]$, where ℓ is determined by a "cushion" hyperparameter [1]. On singular values, one step of Muon+PE applies a scalar polynomial

$$p_{\mathrm{PE}}(\sigma) \;=\; a\,\sigma \;+\; b\,\sigma^3 \;+\; c\,\sigma^5, \tag{6}$$

which maps σ toward 1 when $\sigma \in [\ell, 1]$.

Lifting (6) back to matrices, a single Muon+PE step can be written as

$$X_{k+1} \;=\; a\,X_k \;+\; b\,X_k X_k^\top X_k \;+\; c\,X_k\big(X_k^\top X_k\big)^2, \tag{7}$$

since $X_k X_k^\top X_k$ and $X_k(X_k^\top X_k)^2$ correspond to the σ^3 and σ^5 terms in (6) when $X_k = U\Sigma_k V^\top$.

Concrete coefficients used in our Muon+PE runs. In our experiments we use the standard degree-5 polynomial sign/polar approximant from the reference implementation (often attributed to Keller et al. [5]), with

$$(a, b, c) \;=\; (3.4445,\; -4.7750,\; 2.0315). \tag{8}$$

On singular values this corresponds to

$$p_{\mathrm{PE}}(\sigma) = 3.4445\,\sigma - 4.7750\,\sigma^3 + 2.0315\,\sigma^5,$$

and at the matrix level we implement (7) with these (a, b, c) values wherever Muon+PE is applied (all matrices or VO/FFN-only, depending on the configuration in Table 3). We do not re-derive these coefficients; we simply reuse the standard polynomial from the underlying library and treat the iteration budget and cushion as hyperparameters in Phase 1.

B Polar Express hyperparameters

The original Polar Express paper [1] fixes the cushion parameter to a specific value, $c = 0.02407327424182761$. This constant is obtained numerically in their polynomial-design pipeline, chosen so that the minimax polynomial remains stable over a target spectral range (and in low precision); no closed-form derivation is given. In our experiments we therefore treat $c \approx 0.024$ as the paper default and sweep nearby values (e.g., $c \in \{0.024, 0.04\}$) together with the iteration budget T to study how sensitive Muon+PE is to this choice.

C CASPER Compute Architecture

All experiments were conducted on hardware generously provided by the Collaboration for Astronomy Signal Processing and Electronics Research (CASPER), funded by the National Science Foundation (NSF). This self-administered infrastructure consists of two high-performance servers, one with two NVIDIA RTX A6000 GPUs and another with NVIDIA RTX 4070/5070 GPUs.

To maximize efficiency, the training pipeline employed a tiered data loading strategy, where active data were streamed from local 1 TB SSDs using page-pinned memory and archival data was stored on a 12 TiB BeeGFS filesystem. This system enabled the A6000s to each maintain a $\approx$ 92k tokens/sec throughput and 100% GPU Streaming Multiprocessor (SM) utilization during multi-hour training runs.

A Raspberry Pi 5 (`radiopi5`) acts as the gateway server to a secure internal network, within which users can manage jobs on two heterogeneous compute servers (`a6k`, `gf`), networked via a 100 Gbps Ethernet interconnect. Figure 8 visualizes this system and Table 1 provides detailed GPU-node specifications.

C.1 Detailed Specifications

Head Node: Command and control is managed via a Raspberry Pi 5 (`radiopi5`) using Raspberry Pi Connect/VNC.

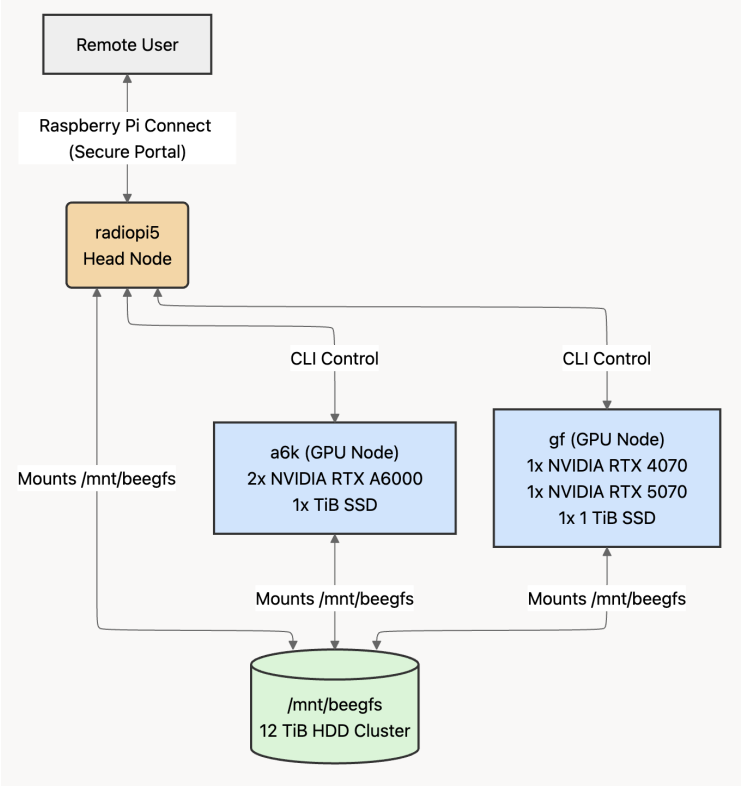

Figure 8: HPC topology: A Raspberry Pi 5 head node and two GPU compute nodes are networked together and mount the common BeeGFS volume `/mnt/beegfs` for file sharing.

Table 1: Detailed Compute Node Specifications

Specification	**Compute Node A** (`a6k`)	**Compute Node B** (`gf`)
Operating System (OS)	Ubuntu 22.04.5 LTS	Ubuntu 22.04.5 LTS
CPU	Intel Xeon Silver 4410T	Intel Xeon Silver 4410T
GPU0	NVIDIA RTX A6000 (48GB Ampere)	NVIDIA RTX 5070
GPU1	NVIDIA RTX A6000 (48GB Ampere)	NVIDIA RTX 4070
Local Storage (Hot Tier)	1 TB SSD	1 TB SSD

Storage Tiering: A dual-tier system is employed: the local SSDs serve as the **Hot Tier** for active dataset shards and immediate checkpoints, and a 3-server BeeGFS (v8.1.0) parallel file system (12 TB capacity) serves as the **Cold Tier** for log aggregation and checkpoint archiving, mounted at `/mnt/beegfs`.

D Training Recipe & Reproducing Results

To reproduce our training pipeline and re-create our plots, follow the `README.md` at our public GitHub repo: https://github.com/nraultwang/cs182-project/tree/main. Any CPU platform should be sufficient to re-create our main figures. However, to successfully execute our training pipeline to reproduce ALL of our sweeps, you will need to partially replicate our hardware configuration specified in Appendix C. At a minimum you will need an SSD with at least 250 GB of free space for checkpointing and access to an NVIDIA GPU with 1) at least 32 GB of VRAM and 2) `bfloat16` support.

Please consult Table 2 for details regarding our training recipe.

Table 2: Fixed Training Recipe Hyperparameters (GPT-2 Small)

Parameter	Value / Setting	Notes
Model Architecture	GPT-2 Small	12-Layer, 768 Hidden Dim, 12 Heads
Data Set	FineWeb	1 Billion tokens; follows [1]
Precision	bfloat16	Polar Express baseline; follows [1] Context Length
1024	Tokens per sequence; follows [1]	
Batch Size (Global)	512	Sequences per optimizer step; follows [1]
Mini-batch Size	32	Sequences per gradient accumulation step
GPU Hardware	A6000 (48GB Ampere)	Data in Sec. 5 collected by server A (`a6k`)
Initialization		
Linear Weight Init (1/2)	$\mathrm{N}(0, 0.02^2)$	GPT-2 paper default for linear layers
Linear Weight Init (2/2)	$\mathcal{N}(0, (0.02/\sqrt{2 \times \texttt{n\_layer}})^2)$	Layers with `LLMC_RESIDUAL_SCALE_FLAG=true`
Embedding Weight Init	$\mathrm{N}(0, 0.02^2)$	GPT-2 paper default
Bias Init	0	GPT-2 paper default
Training Schedule		
Phase 1 Epoch size	333M FineWeb tokens	Phase 1 (Initial stability/convergence)
Phase 2 Epoch size	1B FineWeb tokens	Used for Phase 2 final accuracy/loss evaluation
Learning Rate Schedule	Constant Linear	0.4 fraction warmup; follows [1]
Random seeds	42, 123, 456	All runs repeated 3 times, once for each seed.

E Experimental Protocols

This appendix details the specific procedural steps and sweep parameters used for the experiments described in Section 4.

E.1 Phase 1 Protocol

To isolate effects one at a time, we first probe the *safety factor*, then sweep the *iteration budget/schedule*, and finally sweep the *cushion* at fixed iteration budgets selected from the iteration sweep.

- **Safety factor.** We vary
$$\texttt{safety factor} \in \{1.00, 1.01\}$$
at fixed iteration budget and cushion to verify that small changes in safety factor do not materially affect stability, orthogonality error, or loss.
- **Iteration budget and schedule.** We vary the number of iterations to study the effect of iteration count and simple schedules:
  - single-budget settings such as $[3]$, $[5]$, $[7]$;
  - short schedules such as $[3, 0]$, $[5, 1]$, and $[2]$, where a zero denotes "no PE step" for that position, yielding periodic or mixed-strength polar updates.

  Lists are cycled over training steps (e.g., $[5, 1]$ alternates strong and weaker steps; $[5, 0]$ is block-periodic).
- **Cushion sweep.** Using `phase1-cushion-sweep`, we fix the number of iterations to two representative budgets selected from the iteration sweep (e.g., $[5]$ and $[3]$) and sweep
$$\texttt{cushion} \in \{0.015, 0.020, 0.024, 0.030, 0.035, 0.04\},$$
with degree fixed at 5 and safety factor near paper defaults. This isolates the cushion's effect on spectra/orthogonality conditional on fixed iteration budgets.

E.2 Phase 2 Protocol

For each optimizer—AdamW, Muon+NS(Full), Muon+PE(Full), Muon+NS(VO/FFN), Muon+PE(VO/FFN)—we run a short 0.3-epoch learning-rate sweep over a small grid around a plausible LR scale.

From this sweep we select a single *baseline* LR per optimizer as the LR that minimizes final train loss, and we use that LR for the 1-epoch loss comparisons in Section 5. Using the same sweeps, we then examine attention-entropy trajectories at the baseline and all larger LRs, and define an entropy-based maximum stable learning rate (MSLR) as the largest LR for which none of the monitored layers satisfy the collapse criterion in Appendix F. Full LR grids, numerical results, and a detailed discussion of the relationship between baseline LR and MSLR are deferred to Appendix I.

F Evaluation Metrics

To rigorously evaluate the stability and optimization quality of Polar Express compared to standard baselines, we track end-to-end training metrics (loss, perplexity, throughput) together with a set of internal diagnostics that target orthogonality, singular value spectra, attention health, and scale.

PE Orthogonality Error. Since the core premise of MUON and Polar Express is to enforce (approximate) orthogonality in the update steps, we directly measure how close the update matrices are to being orthogonal. Let G_t denote a raw gradient update and $\tilde{G}_t$ the corresponding post-PE update. For a generic matrix X, we define the orthogonality error

$$\varepsilon(X) = \left\| X^\top X - I \right\|_F. \tag{9}$$

During training, the optimizer logs $\varepsilon_{\text{before}} = \varepsilon(G_t)$ and $\varepsilon_{\text{after}} = \varepsilon(\tilde{G}_t)$ for a stream of sampled updates, which we aggregate into the scalar metrics `pe/ortho_err_before` and `pe/ortho_err_after` (averaged over a small temporal window). In addition, we record per-layer orthogonality errors for sentinel attention layers (layers 0, 5, and 11) in the stacked-QKV setting, yielding metrics such as `ortho_err_before/layer0_stacked_qkv`. These quantities quantify how strongly Polar Express reshapes $X^\top X$ toward the identity and how this effect varies across depth.

Singular Value Diagnostics. To go beyond a single Frobenius-norm scalar, we periodically compute singular value decompositions for cached post-PE update matrices in selected layers. If $U \in \mathbb{R}^{m \times n}$ is a post-PE update with singular values $\sigma_1 \geq \cdots \geq \sigma_r$, we log histograms and summary statistics of $\{\sigma_i\}$ (including, e.g., $\sigma_{\max}$, $\sigma_{\min}$, effective rank, and simple spectral gaps) for stacked QKV as well as the split Q, K, and V blocks and the VO/FFN matrices. These SVD-based metrics are logged at a lower frequency due to their cost, and they allow us to track how the singular value distributions of the updates evolve over training under different Polar Express iteration budgets, cushions, application modes, and VO/FFN vs. full-layer deployment.

Attention Health Metrics. Following Zhai et al. [13], we monitor attention behavior as a proxy for training stability and potential attention-entropy collapse. For checkpoint layers (again layers 0, 5, and 11), we reconstruct the pre-softmax attention logits and post-softmax attention probabilities for each head. For a given head h and query position i, with attention distribution $A_{h,i} \in \mathbb{R}^N$ over N keys, we define the entropy

$$H(A_{h,i}) = -\sum_{j=1}^{N} A_{h,i,j} \log\big(A_{h,i,j} + 1\text{e}{-10}\big). \tag{10}$$

We log the mean attention entropy and its lower/upper quantiles (e.g., 5th and 95th percentiles) across heads, tokens, and a subsampled set of positions, as well as the fraction of queries whose maximum attention weight exceeds a high threshold such as 0.95:

$$\text{frac\_maxA}{>}0.95 = \frac{1}{BTH} \sum_{b,t,h} \mathbf{1}\Big[\max_j A_{h,(b,t),j} > 0.95\Big]. \tag{11}$$

We also record basic statistics of the unmasked logits (mean, standard deviation, and a high quantile), which serve as a check on the scale of the attention scores. Together, these metrics quantify whether Polar Express hyperparameters and MUON deployment choices (full vs. VO/FFN-only, stacked vs. split, per-head splits) correlate with healthy, high-entropy attention or with premature collapse.

Attention-entropy collapse criterion. For each sentinel layer $\ell \in \{0, 5, 11\}$ and learning rate α, we aggregate the logged mean attention entropy across seeds at the final logged step. Let $\mu_{\ell,\alpha}$ denote this across-seed mean and $\mathrm{SEM}_{\ell,\alpha}$ the corresponding standard error. We define the lower bound of the 95% confidence interval as

$$\mathrm{LCB}_{\ell,\alpha} = \mu_{\ell,\alpha} - 1.96 \cdot \mathrm{SEM}_{\ell,\alpha} .$$

We say that layer ℓ has *attention-entropy collapse* at learning rate α if $\mathrm{LCB}_{\ell,\alpha} \leq 0.5$; otherwise we treat the layer as *stable*. This binary notion of collapse is what we use to report maximum stable learning rates and layerwise stability patterns in Section 5.

Scale and Gradient-Flow Metrics. Finally, we track norms of key weights and gradients to monitor scale and gradient flow. For attention layers at depths 0, 5, and 11, we split the stacked QKV weight matrix into Q, K, and V blocks and log average row norms (e.g., `qkv/layer0/q_norm/mean`, `qkv/layer0/k_norm/mean`, `qkv/layer0/v_norm/mean`). We also record overall weight norms for the attention output (W_O) and FFN blocks in these layers, along with per-layer gradient norms and subpath gradient norms for the MLP-up projection and attention projections. These metrics help detect vanishing or exploding gradients and large shifts in weight scale that might interact with the orthogonalization step.

In combination, the orthogonality errors, singular value diagnostics, and attention/scale metrics provide a detailed view of how Polar Express changes the spectrum of updates and the internal dynamics of the Transformer, and how selective VO/FFN deployment compares to full-layer and AdamW baselines, beyond what can be seen from loss and perplexity alone.

G Optimizers and Baselines

Table 3: Summary of Optimizer Variants

Abbreviation	Orthogonalization Method	Muon-updated Matrices	Paper
Muon+PE(Full)	Muon Polar Express	QKV, O, FFN	Amsel et al. [1]
Muon+PE(VO/FFN)	Muon Polar Express	V, O, FNN	Amsel et al. [1]
Muon+NS(Full)	Muon Newton-Schulz	QKV, O, FFN	Keller et al. [5]
Muon+NS(VO/FFN)	Muon Newton-Schulz	V, O, FNN	Keller et al. [5]
AdamW	AdamW	N/A	N/A

Dedicated hyperparameter sweeps for learning rate and weight decay were conducted for each of the five optimizer variants specified in Table 3. This calibration enables fair comparisons between the optimizers, mitigating the possibility that results presented in Sec. 5 are confounded with poorly-chosen hyperparameters.

Weight decay. Unless otherwise noted, all optimizers use decoupled weight decay 0.01 on the parameters they update (AdamW-only parameters as well as Muon-updated matrices). The only exception is the Phase 2 full-epoch comparison runs in Figure 7, for which we set weight decay to 0 for all optimizers in order to isolate the effect of the learning rate and orthogonalization scheme.

H Additional Phase 2 Ablations

This section summarizes two exploratory Phase 2 ablations that are not part of the core optimizer comparison: (i) a "cheap" Muon+PE configuration with reduced Polar Express iteration count, and (ii) a split-QKV Muon+PE configuration that orthogonalizes Q, K, and V separately rather than as a single stacked matrix.

H.1 Baseline vs. cheap Muon+PE

Motivated by the Phase 1 cushion and iteration study (Section 6.1), we compare a baseline Muon+PE configuration using $T = 5$, $c = 0.024$ to a cheaper variant using $T = 3$, $c = 0.04$ in full 1-epoch

GPT-2 Small runs. Both use the same VO/FFN and QKV/O placement as the standard Muon+PE configuration, and each is run at its loss-tuned baseline learning rate from the corresponding 0.3-epoch sweep.

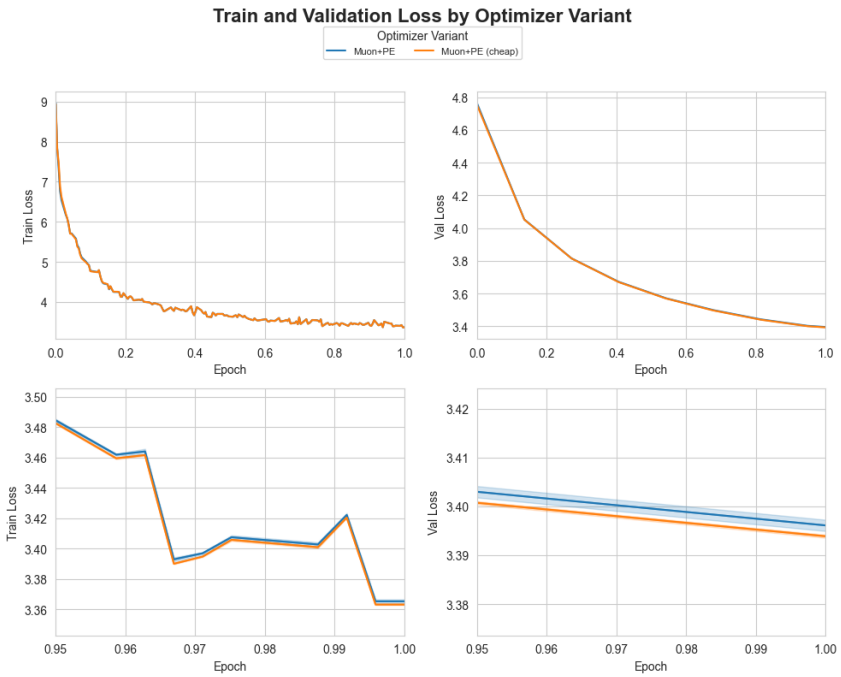

Figure 9: Train and validation loss for the baseline Muon+PE configuration ($T = 5$, $c = 0.024$) versus a cheaper Muon+PE configuration ($T = 3$, $c = 0.04$) in 1-epoch GPT-2 Small runs. Loss curves are nearly indistinguishable, consistent with Phase 1 results indicating that a modestly larger cushion can compensate for a smaller iteration budget in this regime.

Across the three seeds used in our Phase 2 setup, the loss trajectories of the baseline and cheap Muon+PE variants are very similar, with final train and validation loss differing only slightly and without a consistent ordering across seeds. Combined with the orthogonality and spectral diagnostics from Phase 1, this supports the view that moderate reductions in T can be traded for a slightly more aggressive cushion without materially degrading optimization quality at GPT-2 Small scale, although the wall-clock timing differences are too noisy to support firm conclusions about end-to-end speed (see Appendix J).

Taken together with the Phase 1 0.3-epoch sweeps, these results suggest a more nuanced picture. In the short-horizon sweeps, the baseline configuration with $T = 5$, $c = 0.024$ slightly outperforms the cheaper $T = 3$, $c = 0.04$ setting, whereas over a full epoch the cheap configuration very slightly pulls ahead in final loss. One possible interpretation in the SVD view is that the baseline update preserves more anisotropy early in training, allowing gradients to move more directly along a few dominant directions, while the cheaper configuration, with its stronger spectrum-flattening, yields more isotropic updates that may better preserve a broader set of directions later in training. Under this hypothesis, early phases of training benefit from moderately anisotropic updates, whereas later phases may prefer more isotropic updates that avoid "zeroing out" low-singular-value directions. Given the small effect sizes and limited number of seeds, these patterns should be viewed as suggestive rather than conclusive.

H.2 Split-QKV Muon+PE

We also consider a split-QKV Muon+PE configuration that applies Muon separately to the Q, K, and V projections in attention, rather than to a single stacked $[Q; K; V]$ matrix. This variant uses the same Polar Express hyperparameters as the baseline Muon+PE configuration and the same placement on the output and FFN matrices.

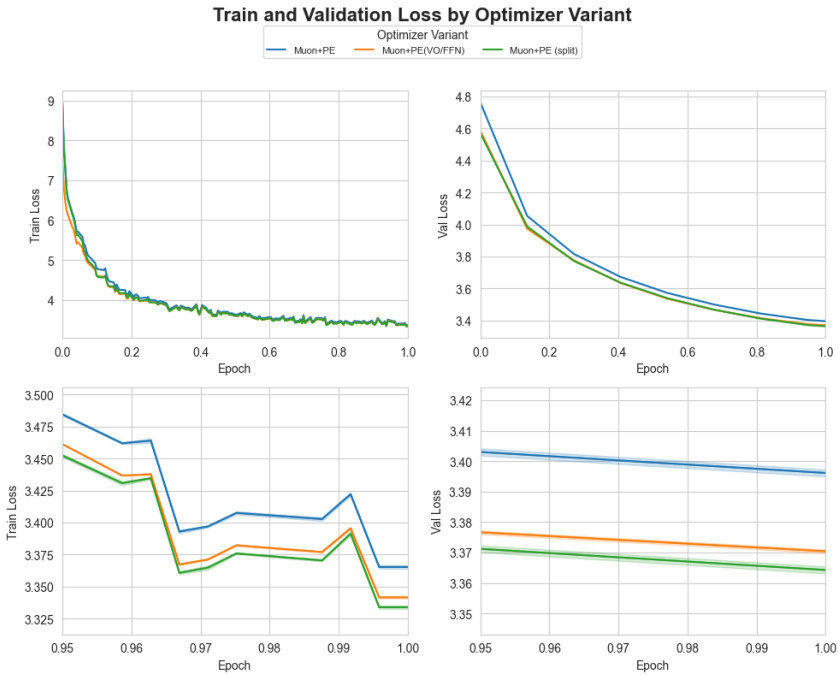

Figure 10: Train and validation loss for Muon+PE (stacked QKV), Muon+PE(VO/FFN), and split-QKV Muon+PE. All runs use 1-epoch GPT-2 Small training at their respective loss-tuned baseline learning rates.

In these exploratory runs, the split-QKV Muon+PE configuration achieves slightly lower final validation loss than both the stacked-QKV Muon+PE baseline and Muon+PE(VO/FFN), while remaining within the same overall stability regime. Combined with the observation that Muon+PE(VO/FFN) itself slightly outperforms the all-matrices Muon+PE configuration, this pattern is directionally consistent with prior work suggesting that VO/FFN-only Muon recovers most of the benefit of full-layer Muon [12], and may even hint at an additional advantage from decoupling the QKV block. A plausible hypothesis is that independently orthogonalizing (or not orthogonalizing) Q, K, and V allows these projections to specialize more freely, whereas enforcing a single stacked polar step on $[Q; K; V]$ imposes a stronger geometric constraint that can slightly hinder such specialization. However, the observed gains are modest and based on a small number of seeds, so they are best interpreted as suggestive evidence that QKV splitting deserves further study rather than as a definitive conclusion.

I Learning-rate Calibration and MSLR

For each optimizer in Table 3 we ran short 0.3-epoch learning-rate (LR) sweeps as described in Section 4.2. These sweeps serve two roles. First, they provide a *baseline* LR for each optimizer: the LR that minimizes final train loss at 0.3 epochs. Second, using the attention-entropy metrics and collapse criterion from Appendix F, we reuse the same sweeps to determine an entropy-based maximum stable learning rate (MSLR) by scanning to larger LRs and checking when any of the monitored layers (0, 5, 11) collapses.

Table 4 summarizes the resulting baseline LRs and MSLRs, together with the ratio between them.

The AdamW and all-matrices Muon variants (Muon+NS and Muon+PE on QKV/O/FFN) admit several-fold LR increases beyond their loss-optimal baselines before any monitored layer violates the attention-entropy collapse criterion, yielding MSLR multipliers of roughly $4\times$ and $5\times$, respectively. By contrast, the VO/FFN-only Muon variants have a much narrower stability window: their MSLR is essentially equal to their baseline LR, and even modest LR increases beyond this point trigger attention-entropy collapse, typically beginning in the earliest layer. Within each placement

Table 4: Baseline learning rates (chosen by final train loss on 0.3-epoch sweeps) and entropy-based maximum stable learning rates (MSLR) for each optimizer. Multipliers are relative to the baseline LR. Muon+PE-mod is included in the 1-epoch loss comparison but was not part of the Phase 2 entropy sweep.

Optimizer	Baseline LR (train loss)	MSLR (entropy)	MSLR / Baseline
AdamW	3.0×10^{-4}	1.2×10^{-3}	$\approx 4\times$
Muon+NS	1.0×10^{-2}	5.0×10^{-2}	$\approx 5\times$
Muon+PE	1.0×10^{-2}	5.0×10^{-2}	$\approx 5\times$
Muon+NS(VO/FFN)	3.0×10^{-3}	3.0×10^{-3}	$\approx 1\times$
Muon+PE(VO/FFN)	3.0×10^{-3}	3.0×10^{-3}	$\approx 1\times$
Muon+PE-mod	1.0×10^{-2}	—	—

(all matrices vs. VO/FFN), the MSLRs for Muon+NS and Muon+PE are nearly identical in our regime.

Non-learning high-LR regime. The refined LR sweeps also include very aggressive LRs that exceed these MSLRs, sometimes by more than an order of magnitude. At such LRs we occasionally observe non-monotonic attention-entropy trajectories in which entropy appears to recover after an initial collapse, but the corresponding train-loss curves remain large and highly noisy. We interpret this "entropy rebound" behavior as a non-learning, high-noise regime where the model bounces around high-loss regions and attention looks high-entropy simply because it is effectively random. For this reason, the main Phase 2 analysis (Section 5.2) and the MSLR values in Table 4 are restricted to the $\sim$ 1–10$\times$ LR window where the models actually learn, and we do not treat entropy rebounds at extreme LRs as evidence of genuine stability.

I.1 Layerwise attention-entropy stability

For each optimizer and learning rate in the Phase 2 sweeps, we mark the monitored layers (0, 5, 11) as *stable* (S) or exhibiting *attention-entropy collapse* (C) under the attention-entropy collapse criterion in Appendix F. These per-optimizer tables underlie the entropy-based MSLRs summarized in Table 4. Rows with LR/baseline substantially larger than $10\times$ fall into the high-LR, non-learning regime described above.

Table 5: Layerwise attention-entropy stability for AdamW. "S" denotes a stable layer (LCB > 0.5) and "C" denotes attention-entropy collapse (LCB ≤ 0.5). Baseline LR is 3×10^{-4}.

LR	LR / Baseline	Layer 0	Layer 5	Layer 11
0.0003	$1\times$	S	S	S
0.0006	$2\times$	S	S	S
0.0009	$3\times$	S	S	S
0.0012	$4\times$	S	S	S
0.0015	$5\times$	S	C	C
0.0018	$6\times$	C	S	C
0.0021	$7\times$	C	C	C
0.0024	$8\times$	C	C	C

We also verified that using stricter attention-entropy thresholds (e.g., LCB ≤ 1 instead of 0.5) leaves the Muon MSLRs unchanged and only reduces the AdamW MSLR modestly (from $\approx 4\times$ to $\approx 3\times$ its baseline LR), so the qualitative ordering of optimizer stability is robust to the exact collapse threshold.

J Wall-clock measurements

We benchmarked wall-clock efficiency by running a single 1-epoch training run for each optimizer variant under a common data pipeline and hardware setup, logging total train time, mean step time,

Table 6: Layerwise attention-entropy stability for Muon+NS (QKV/O/FFN). Baseline LR is 1×10^{-2}.

LR	LR / Baseline	Layer 0	Layer 5	Layer 11
0.010	1×	S	S	S
0.020	2×	S	S	S
0.030	3×	S	S	S
0.040	4×	S	S	S
0.050	5×	S	S	S
0.060	6×	C	S	S
0.070	7×	C	S	S
0.080	8×	C	S	S
0.090	9×	C	C	S
0.100	10×	C	C	S
0.110	11×	C	C	S
0.120	12×	C	S	S
0.130	13×	C	S	S
0.140	14×	C	S	S
0.150	15×	C	S	S
0.160	16×	C	S	S

Table 7: Layerwise attention-entropy stability for Muon+PE (QKV/O/FFN). Baseline LR is 1×10^{-2}.

LR	LR / Baseline	Layer 0	Layer 5	Layer 11
0.010	1×	S	S	S
0.020	2×	S	S	S
0.030	3×	S	S	S
0.040	4×	S	S	S
0.050	5×	S	S	S
0.060	6×	C	S	S
0.070	7×	C	S	S
0.080	8×	C	C	S
0.090	9×	C	C	S
0.100	10×	C	C	S
0.110	11×	C	C	S
0.120	12×	C	C	S
0.130	13×	C	S	S
0.140	14×	C	S	S
0.150	15×	C	S	S
0.160	16×	C	S	S

and (for Muon+PE variants) the average time spent in the Polar Express substep. Table 10 reports these measurements.

Our Phase 1 sweeps varied the Muon-specific hyperparameters that control per-step cost, namely the number of iterations T and the cushion c. Throughout the paper we treat $T = 5$, $c = 0.024$ as our reference Muon configuration, since it is used in most experiments, and compare it to additional runs with $T = 3$ and $T = 6$. A lighter setting with $T = 3$, $c = 0.04$ yields loss curves that are very close to the $T = 5$, $c = 0.024$ baseline at our chosen learning rates while reducing the number of Muon iterations by 40%. A heavier setting with $T = 6$ (keeping $c = 0.024$) gives slightly better orthogonalization and cleaner singular-value behavior at the cost of one extra iteration per step. Because our wall-clock measurements are noisy, we do not claim a single globally optimal choice: if profiling on the target hardware shows that additional Muon iterations cause a clear, repeatable slowdown, then $T = 3$, $c = 0.04$ is a reasonable "fast" configuration that closely matches the $T = 5$, $c = 0.024$ baseline; if iteration count has little effect on wall-clock time, we instead recommend using $T = 6$, $c = 0.024$ by default, taking advantage of the stronger orthogonalization.

Table 8: Layerwise attention-entropy stability for Muon+NS(VO/FFN). Baseline LR is 3×10^{-3}.

LR	LR / Baseline	Layer 0	Layer 5	Layer 11
0.003	$1\times$	S	S	S
0.004	$\approx 1.33\times$	C	S	S
0.005	$\approx 1.67\times$	C	C	S
0.006	$2\times$	C	C	S
0.007	$\approx 2.33\times$	C	C	S
0.008	$\approx 2.67\times$	C	C	S
0.009	$3\times$	C	C	C
0.010	$\approx 3.33\times$	C	C	C

Table 9: Layerwise attention-entropy stability for Muon+PE(VO/FFN). Baseline LR is 3×10^{-3}.

LR	LR / Baseline	Layer 0	Layer 5	Layer 11
0.003	$1\times$	S	S	S
0.004	$\approx 1.33\times$	C	S	S
0.005	$\approx 1.67\times$	C	C	S
0.006	$2\times$	C	C	S
0.007	$\approx 2.33\times$	C	C	S
0.008	$\approx 2.67\times$	C	C	C
0.009	$3\times$	C	C	C
0.010	$\approx 3.33\times$	C	C	C

Additional Phase 2 1-epoch runs directly compare the baseline Muon+PE configuration ($T = 5$, $c = 0.024$) to this lighter "cheap" variant ($T = 3$, $c = 0.04$) at their loss-tuned learning rates. Their train and validation loss curves are nearly indistinguishable (Appendix H, Figure 9), reinforcing the view that moderate reductions in iteration count can be traded for a slightly more aggressive cushion without materially degrading optimization in this regime.

Across AdamW and all Muon variants, the observed differences in both total train time and mean step time are very small (all runs lie within roughly 1% of one another) and not consistently ordered by optimizer. Given that we collected only a single timing run per configuration and that small changes in background load or caching can easily account for such differences, we do not interpret these results as a reliable ranking of wall-clock efficiency. Instead, we treat them as a rough sanity check that the MUON variants do not incur a large, systematic slowdown relative to AdamW, and we base our main conclusions on loss and stability metrics rather than these preliminary timing measurements.

Table 10: Wall-clock statistics for 1-epoch GPT-2 Small runs. Each entry reports the mean across seeds in our timing sweeps.

Variant	Total train time (s)	Step time (ms)	PE step time (ms)
AdamW	9835.8	5660.2	–
Muon+NS (VO/FFN)	9907.7	5689.6	0.6
Muon+NS	9916.8	5688.8	0.6
Muon+PE (VO/FFN)	9931.8	5689.7	0.9
Muon+PE	9912.1	5686.8	0.6
Muon+PE (cheap)	9950.1	5726.8	0.9

K Response to Poster Session and Final Reviewer Feedback

This section outlines the revisions made to the final report in response to the feed back we received during the peer review process. We have addressed concerns regarding terminology, reproducibility, and the scope of our literature review as follows:

K.1 Nomenclatural Consistency and Definitions

We attempted to standardize our nomenclature for "safety factor" and "cushion" to improve readability, as noted by our final report reviewers. We expanded and simplified our optimizer notation definition Table 3 to improve the consistency of our notation for the Muon variants (Newton-Schulz, Polar Express).

K.2 Clarifying Weight Decay

Some reviewers were confused why we used different weight decay values. We clarified that these weight decay = 0 runs were to reproduce a figure from [1]. This data is displayed in our Figure 7.

For all other runs we use weight decay 0.01, as noted in Appendix G.

K.3 Clarifying the Introduction

As suggested by Professor Ranade during the poster presentation session, we modified our abstract and introduction to better highlight and clarify the motivation behind our project and experiments.

K.4 Literature Review Expansion

We expanded our literature review in response to reviewer critiques that we were too narrow in our treatment to adequately situation Polar Express. We have added additional discussion of geometry-based optimizers (Shampoo, MuonBP) and added more motivation on why we chose to focus on the VO/FFN matrices.

See Section 2 for the updated literature.

K.5 Data Visualization and Analysis

To address Reviewer #1's comment about difficult-to-interpret singular value heatmap (originally Figure 6), we have improved the spectral heatmap plot to clarify the effect of the number of Polar Express steps on the singular value spectrum of the momentum matrix. By plotting $T = 3$ vs $T = 5$ instead of $T = 3$ vs $T = 6$, the singular value spectrum is less peaked around a small range of values, resulting in a density color scale is easier to interpret and compare across the side-by-side both plots. See Figure 2 for this new figure.

L Response to Initial Reviewer Feedback

This section outlines the revisions made to the final report in response to the feed back we received during the peer review process. We have addressed concerns regarding terminology, reproducibility, and the scope of our literature review as follows:

L.1 Nomenclatural Consistency and Definitions

Reviewers identified inconsistencies in the naming of our core method (previously referred to interchangeably as "Polar Express," "Muon+PE," and "Keller"). To resolve this, we have standardized all references to "Muon+PE" or "Polar Express" throughout the manuscript. Additionally, we created Fig. 12, an annotated Transformer architectural diagram to clarify what we mean by full-layer versus VO/FFN-only Muon application.

L.2 Experimental Reproducibility and Hyperparameters

We have corrected the description of our experimental infrastructure to accurately reflect the use of Weights & Biases for metrics collection, and clarified our precise "fixed Hydra recipe" in Table 2. This table explicitly enumerates all fixed training parameters—including learning rate schedules, batch sizes, sequence lengths—replacing previous generic descriptions.

To ensure full reproducibility, we have also appended added Appendices G and I to explain how we calibrated and defined our optimizer variants.

L.3 Addressing Confounding Variables

We address the reviewer's concern regarding hardware heterogeneity (e.g., mixing RTX A6000 and RTX 4070 GPUs). To preempt confusion regarding confounding effects from this heterogeneous hardware, we have explicitly clarified that all our data was collected on the A6000 cards.

L.4 Data Visualization and Analysis

We have improved the legibility of all figures, ensuring labels are readable at standard document zoom levels. To better represent statistical uncertainty, all performance curves now include 95% confidence intervals derived from multiple trials. Finally, we have integrated the complete results from Phase 1 (hyperparameter efficiency) and Phase 2 (attention stability), providing a definitive analysis of the "attention hardening" hypothesis and its impact on entropy collapse.

M AI Usage Acknowledgement

We used a variety of Frontier LLM models to accelerate our report drafting, coding, and debugging. Below is a list of models we used for this project:

- Gemini 2.5 Pro
- Gemini 3 Pro
- claude 4.5 sonnet
- GPT5.1 Codex
- GPT5.1 High-Reasoning
- GPT5.1 Low-Reasoning

N Spiritual Inspiration

Figure 11: Train [10]

O Transformer Architecture: Stacked-QKV, Split-QKV

Figure 12 shows which components of a standard Transformer layer are updated by AdamW and the Muon variants. Note that Muon can only update 2-dimensional tensors while AdamW may update both 1-dimensional and 2-dimensional tensors.

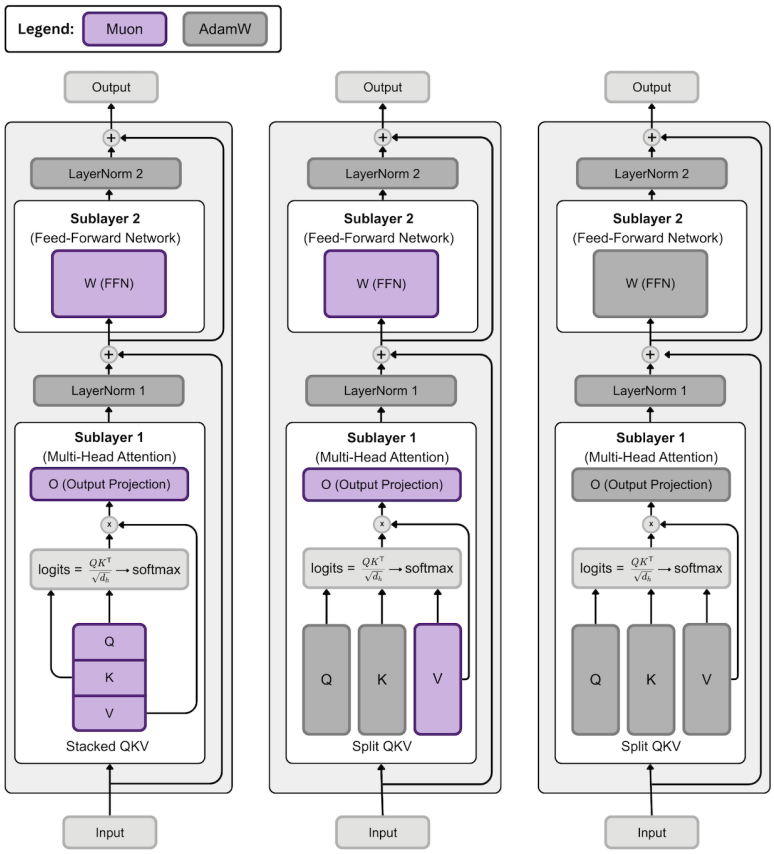

Figure 12: Components of a Transformer layer updated by Muon versus AdamW. Left (”Stacked-QKV”): Muon+PE(Full) and Muon+NS(Full) update the stacked Query-Key-Value, Output, and FFN matrices. Middle (”Split-QKV”): Muon+PE(VO/FFN) and Muon+NS(VO/FFN) update the Value, Output and FFN matrices. Right (”Split-QKV”): AdamW updates all parameters.