TRASMUON: TRUST-REGION ADAPTIVE SCALING FOR ORTHOGONALIZED MOMENTUM OPTIMIZERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Muon-style optimizers leverage Newton-Schulz (NS) iterations to orthogonalize updates, yielding update geometries that often outperform Adam-series methods. However, this orthogonalization discards magnitude information, rendering training sensitive to step-size hyperparameters and vulnerable to high-energy bursts. To mitigate this, we introduce TrasMuon (**T**rust **R**egion **A**daptive **S**caling **Muon**). TrasMuon preserves the near-isometric geometry of Muon while stabilizing magnitudes through (i) global RMS calibration and (ii) energy-based trust-region clipping. We demonstrate that while reintroducing adaptive scaling improves optimization efficiency, it typically exacerbates instability due to high-energy outliers. TrasMuon addresses this by defining a trust region based on relative energy ratios, confining updates to a stable zone. Empirical experiments on vision and language models demonstrate that TrasMuon converges faster than baselines. Furthermore, experiments without warmup stages confirm TrasMuon's superior stability and robustness.

1 INTRODUCTION

Optimizer choice remains a bottleneck for training modern foundation models, shaping convergence and stability at scale (DeepSeek-AI et al., 2025; OpenAI, 2026; Anthropic, 2026). In practice, heterogeneous and heavy-tailed/outlier updates can trigger loss spikes and narrow the stable learning-rate region (Behrouz et al., 2025; Kimi Team et al., 2025; Park et al., 2025). Diagonal adaptive methods (Adam/AdamW and variants) provide robust coordinate-wise magnitude control (Kingma & Ba, 2017; Loshchilov & Hutter, 2019; Pagliardini et al., 2025; Marfinetz, 2025), but do not exploit matrix-level update structure. Muon-style optimizers revisit matrix-structured updates via momentum orthogonalization (Bernstein, 2025; Jordan et al., 2024). However, orthogonalization mainly controls *geometry* and discards magnitude information, increasing sensitivity to step-size/warmup choices and vulnerability to bursty, axis-localized energy spikes (Behrouz et al., 2025; Kimi Team et al., 2025; Park et al., 2025).

We propose **TrasMuon** (**T**rust-**R**egion **A**daptive **S**caling for **Muon**), which preserves Muon-style structured mixing while stabilizing magnitudes via global RMS calibration and feature-wise relative-energy trust-region damping. For a matrix parameter $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, TrasMuon applies

$$\Delta W_t \;=\; -\,\hat{\eta}_t\, O_t^{\text{base}}\, \text{diag}(c_t), \qquad c_t \in [c_{\min}, 1]^{d_{\text{in}}}. \tag{1}$$

Here O_t^{base} is obtained by NS orthogonalization plus lightweight row-wise second-moment scaling (NorMuon-style) (Li et al., 2025). The RMS-calibrated step size $\hat{\eta}_t$ improves cross-layer comparability and reduces step-size sensitivity (Bernstein & Newhouse, 2025; Large et al., 2024). The clipping vector c_t selectively suppresses high-energy feature axes while largely preserving the structured mixing factor, and we stabilize this signal via effective-time (schedule-free) weighting (Defazio et al., 2024).

Contributions: We introduce TRASMUON, which combines Muon-style mixing with feature-wise trust-region clipping. TrasMuon achieves faster early-stage convergence and improved stability, exhibiting reduced sensitivity to the learning-rate magnitude and scheduling choices. In particular, it consistently mitigates loss spikes induced by heavy-tailed, axis-localized gradient bursts, while preserving the characteristic geometry of the Muon-style optimizers.

2 Related Work

Diagonal preconditioning and Adam-style optimizers. Adam/AdamW and many refinements remain default baselines due to robust diagonal second-moment scaling (Kingma & Ba, 2017; Loshchilov & Hutter, 2019; Yuan et al., 2025; Pagliardini et al., 2025; Marfinetz, 2025; Shao et al., 2025; Gupta & Wojtowytsch, 2025). These methods stabilize training via coordinate-wise magnitude control but do not explicitly exploit matrix-level structure; TrasMuon instead keeps matrix-structured mixing and adds trust-region adaptive scaling.

Beyond diagonal: block/matrix preconditioners and trust ratios. Richer preconditioners (e.g., K-FAC, Shampoo, Adafactor) capture non-diagonal curvature structure with tractable approximations (Martens & Grosse, 2015; Gupta et al., 2018; Shazeer & Stern, 2018). Layerwise norm/trust-ratio scaling such as LARS/LAMB controls step magnitudes by comparing parameter and update norms (You et al., 2017; 2020). TrasMuon does not estimate curvature factors; it constructs a near-isometric mixing factor via orthogonalization and stabilizes magnitudes using RMS calibration plus a trust-region clipping.

Orthogonalized directions and Muon-style updates. Muon-style optimizers use Newton–Schulz iterations to approximate polar factors, yielding near-isometric directions that can improve Transformer training (Jordan et al., 2024; Bernstein, 2025). Related perspectives connect orthogonalized updates to modular/geometry-aware optimization and practical variants (Bernstein & Newhouse, 2025; Large et al., 2024; Pethick et al., 2025; Ahn et al., 2025; Kumar et al., 2025; Khaled et al., 2025; Riabinin et al., 2025; Li et al., 2025). TrasMuon builds on these directions but targets a distinct failure mode: bursty, axis-localized energy spikes, addressed via feature-wise relative-energy clipping and temporal smoothing.

3 Methodology

We propose **TrasMuon** (**T**rust-**R**egion **A**daptive **S**caling for **Muon**), which *factorizes* matrix updates into a structured mixing factor(illustrated in Appendix B) and feature-wise trust-region adaptive scaling(in Algorithm 1).

Trust-region adaptive scaling. We control bursty, axis-localized updates using the per-column relative-energy ratio $r_{t,j} = E_{t,j}/(E_t^{\mathrm{ref}} + \epsilon)$ and apply multiplicative damping, corresponding to an implicit trust-region constraint $r_{t,j} \lesssim \tau$ (tuned by α and optional trigger k). We measure column energy on pre-orthogonalization momentum M_t,

$$E_{t,j} = \sum_{i=1}^{d_{\mathrm{out}}} M_{t,ij}^2, \tag{2}$$

use a robust reference based on the median,

$$E_t^{\mathrm{cur}} = \mathrm{Quantile}_{0.5}(\{E_{t,j}\}), \tag{3}$$

$$E_t^{\mathrm{ref}} = \beta_E E_{t-1}^{\mathrm{ref}} + (1-\beta_E) E_t^{\mathrm{cur}}, \tag{4}$$

which resists inflation by sparse bursts (Hampel et al., 1986; Huber, 1981). We then apply a smooth, damping-only clip

$$c_{t,j}^{\mathrm{raw}} = \frac{1}{1 + \alpha \log(1 + r_{t,j})}, \tag{5}$$

$$c_{t,j}^{\mathrm{clip}} = \mathrm{clip}\big(c_{t,j}^{\mathrm{raw}}, c_{\min}, 1\big), \tag{6}$$

(optionally only when $r_{t,j} > k$), and set c_t by temporal smoothing (Appendix B).

Schedule-free temporal smoothing. We smooth the instantaneous clip with EMA

$$c_t^{\mathrm{ema}} = \beta_c c_{t-1}^{\mathrm{ema}} + (1-\beta_c) c_t^{\mathrm{inst}}, \tag{7}$$

and optionally apply schedule-free averaging (Defazio et al., 2024) (default $\gamma_t = \eta$):

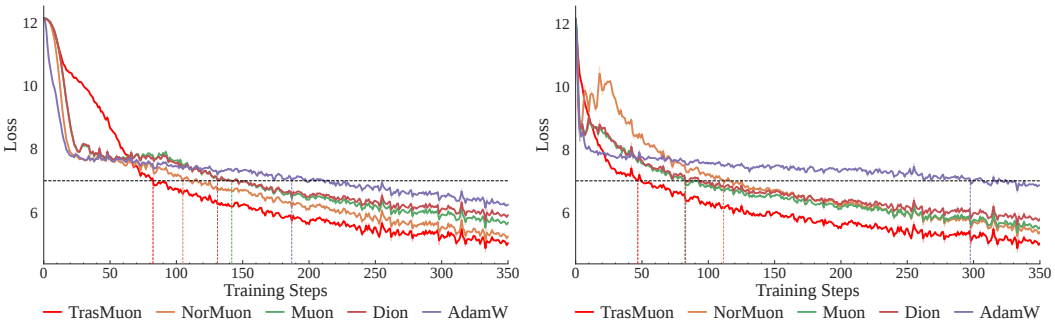
$$S_t = S_{t-1} + \gamma_t^2, \qquad C_t = C_{t-1} + \gamma_t^2 c_{t-1}^{\mathrm{last}}, \tag{8}$$

$$c_t^{\text{avg}} = \frac{C_t}{S_t + \epsilon}, \qquad c_t = (1-\rho)c_t^{\text{ema}} + \rho c_t^{\text{avg}}. \tag{9}$$

We cache $c_t^{\text{last}} \leftarrow c_t$ between gate updates to avoid bias when the raw clip is computed every K steps. Convergence analysis can be seen in Appendix C.

4 EXPERIMENTS

TRASMUON is evaluated across four complementary settings: language-model pretraining under a fixed-budget protocol that highlights early- and late-stage dynamics and includes a minimal replication check, with full hyperparameters and additional plots reported in Appendix D; vision transformers trained with a standard ViT recipe on ImageNet-100 under multi-seed evaluation, together with an additional column-burst stress test, detailed in Appendix E; physics-informed neural networks for Helmholtz under controlled random-ROI sampling shifts that induce reproducible nonstationarity, alongside step-size alignment and learning-rate sensitivity diagnostics in Appendix F; and a controlled diagnostics study with column-localized outlier injection designed to probe the energy-indexed, feature-wise clipping mechanism and its boundary conditions, documented in Appendix G.



(a) Qwen3-0.6B with warmup

(b) Qwen3-0.6B without warmup

Figure 1: Early-stage training dynamics for Qwen3-0.6B from scratch (steps 0–350) under a warmup-stable-decay schedule, comparing (a) warmup-enabled runs and (b) warmup-free runs. The curves are smoothed using a time-weighted exponential moving average (EMA) with smoothing factor 0.1 for better visualization.

Training with warmup. Under the warmup-enabled schedule, all evaluated optimizers exhibit stable training in this fixed-hyperparameter pilot. Although TRASMUON is not the fastest within the first few dozen steps, it begins to reduce loss more rapidly after approximately 80 steps (illustrated in Fig. 1a). At a reference threshold of training loss (7.0), TRASMUON reaches the target in 80 steps, compared to 188 steps for AdamW (2.35×) and 140 steps for Muon (1.75×).[1]

Training without warmup. Without warmup, optimizer becomes more sensitive to step-size calibration. Under the same shared learning rate and batch/sequence configuration, TRASMUON maintains a smooth loss trajectory in the early stage, while several baselines show larger loss oscillations as shown in Fig. 1b. At the same reference threshold of training loss, TRASMUON reaches the target in 48 steps, versus 298 for AdamW (6.21×) and 83 for Muon (1.73×). Despite the widespread use of warmup in pretraining, this setting remains meaningful, as the choice of warmup length is typically determined heuristically.

Late-stage behavior under a fixed budget. Beyond early-stage speed, TRASMUON also attains the lowest (or comparable-lowest) training loss in the late-stage window under both warmup settings in this pilot run. Late-stage curves and loss comparisons are reported in Fig. 2 of Appendix D.2. The early-stage gap narrows as training progresses into a slower-loss regime, suggesting that the benefit is strongest when training dynamics are most nonstationary.

[1] We use loss = 7.0 as a representative checkpoint.

Decreasing feature-wise energy concentration over training. A plausible explanation is that early training exhibits stronger feature-wise anisotropy, where a small subset of hidden dimensions (columns) contributes disproportionately to gradient or momentum energy. In this regime, TRASMUON's energy-based feature-wise clipping is engaged, selectively damping bursty columns and improving stability without discarding the structured Muon direction. As representations become better calibrated, energy may become more uniformly distributed across feature axes, reducing the prevalence of strongly concentrated column-localized bursts; consequently, the clipping signal weakens and the effective update becomes closer to the NorMuon backbone. This explanation is offered as a hypothesis and a direct characterization of activation/gradient anisotropy (e.g., via tracked energy ratios and gate statistics over time) is left to future work.

5 DISCUSSION AND LIMITATIONS

What TRASMUON changes. TRASMUON factorizes matrix updates into (i) a Muon-style near-isometric mixing factor constructed by Newton–Schulz orthogonalization and (ii) explicit magnitude controls: a global RMS-calibrated step size and a bounded, damping-only feature-wise clipping $c_{t,j} \in [c_{\min}, 1]$. This design targets a common practical tension: structured mixing can improve optimization geometry, while stable magnitudes govern learning-rate sensitivity and robustness to heavy-tailed bursts. When feature axes are semantically meaningful and bursts are axis-localized, TRASMUON selectively attenuates high-energy columns while largely preserving the Muon-style mixing structure.

When feature-wise clipping and effective-time smoothing help. Feature-wise clipping is most beneficial when update energy concentrates on a small subset of feature axes, reflected by large relative ratios $r_{t,j} = E_{t,j}/(E_t^{\mathrm{ref}} + \epsilon)$. In this regime, multiplicative clipping suppresses burst-dominated columns without amplification. When clipping is recomputed sparsely (every K steps) or schedules vary, effective-time (schedule-free) averaging provides a stable long-horizon estimate by accumulating γ_t^2-weighted statistics, reducing sensitivity to recomputation frequency and schedule details.

Limitations. (i) The formulation is most natural for 2D weight matrices; extending energy diagnostics and damping to embeddings and higher-order tensors requires careful axis conventions. (ii) Newton–Schulz orthogonalization is sensitive to numerical precision; large-scale deployment benefits from precision-aware implementations. (iii) TRASMUON introduces additional hyperparameters and design choices (e.g., $K, \alpha, k, c_{\min}, \rho$ and the robust reference), and their interactions with model scale and data regimes merit broader sweeps.

6 CONCLUSION

We presented TRASMUON, a Muon-style optimizer that combines (i) NS-based near-isometric mixing factors with (ii) explicit magnitude stabilization via global RMS calibration and bounded, feature-wise trust-region clipping, optionally smoothed by effective-time (schedule-free) averaging. Across the evaluated workloads, TRASMUON improves training stability and achieves competitive or better final performance than strong baselines. Controlled diagnostics further support the intended mechanism: column-localized energy bursts increase relative energy ratios and lead to stronger applied damping, while ablations (e.g., NOCLIP) and broken-axis settings help rule out trivial explanations such as uniform step-size reduction. On practical tasks including language-model training, vision transformers, and PINNs under ROI-induced sampling shifts, TRASMUON yields faster or more stable optimization dynamics and improved robustness(Future Work can be deferred in Appencix A).

IMPACT STATEMENT

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

REFERENCES

Kwangjun Ahn, Byron Xu, Natalie Abreu, Ying Fan, Gagik Magakyan, Pratyusha Sharma, Zheng Zhan, and John Langford. Dion: Distributed orthonormalized updates. *arXiv preprint arXiv:2504.05295*, 2025.

Ambityga. ImageNet100. `https://www.kaggle.com/datasets/ambityga/imagenet100`, 2021. Accessed: 2026-01-23.

Anthropic. Introducing Claude Opus 4.5, 2026. URL `https://www.anthropic.com/news/claude-opus-4-5`. Accessed: 2026-01-12.

Ali Behrouz, Zeman Li, Praneeth Kacham, Majid Daliri, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. ATLAS: Learning to optimally memorize the context at test time. *arXiv preprint arXiv:2505.23735*, 2025.

Jeremy Bernstein. Deriving Muon, 2025. URL `https://jeremybernste.in/writing/deriving-muon`.

Jeremy Bernstein and Laker Newhouse. Modular duality in deep learning. In *International Conference on Machine Learning*, 2025.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, et al. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2025.

Aaron Defazio, Xingyu Alice Yang, Ahmed Khaled, Konstantin Mishchenko, Harsh Mehta, and Ashok Cutkosky. The road less scheduled. In *Advances in Neural Information Processing Systems*, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Zhiwei Gao, Liang Yan, and Tao Zhou. Failure-informed adaptive sampling for PINNs. *SIAM Journal on Scientific Computing*, 45(4):A1971–A1994, 2023.

Kanan Gupta and Stephan Wojtowytsch. Nesterov acceleration in benignly non-convex landscapes. In *The International Conference on Learning Representations*, 2025.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, 2018.

Frank R. Hampel, Elvezio M. Ronchetti, Peter J. Rousseeuw, and Werner A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons, 1986.

Peter J. Huber. *Robust Statistics*. John Wiley & Sons, 1981.

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/`.

Ahmed Khaled, Kaan Ozkara, Tao Yu, Mingyi Hong, and Youngsuk Park. MuonBP: Faster Muon via block-periodic orthogonalization. *arXiv preprint arXiv:2510.16981*, 2025.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, et al. Kimi K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.

Bhavesh Kumar, Roger Jin, and Jeffrey Quesnelle. CurvaDion: Curvature-adaptive distributed orthonormalization. *arXiv preprint arXiv:2512.13728*, 2025.

Tim Large, Yang Liu, Minyoung Huh, Hyojin Bahng, Phillip Isola, and Jeremy Bernstein. Scalable optimization in the modular norm. In *Advances in Neural Information Processing Systems*, 2024.

Zichong Li, Liming Liu, Chen Liang, Weizhu Chen, and Tuo Zhao. NorMuon: Making Muon more efficient and scalable. *arXiv preprint arXiv:2510.05491*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. FineWeb-Edu: The finest collection of educational content, 2024. URL `https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu`. Accessed: 2026-01-01.

Mitchell Marfinetz. Evolving Deep Learning Optimizers. *arXiv preprint arXiv:2512.11853*, 2025.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, 2015.

OpenAI. Introducing GPT-5.2, 2026. URL `https://openai.com/index/introducing-gpt-5-2/`. Accessed: 2026-01-12.

Matteo Pagliardini, Pierre Ablin, and David Grangier. The AdEMAMix optimizer: Better, faster, older. In *International Conference on Learning Representations*, 2025.

Jungwoo Park, Taewhoo Lee, Chanwoong Yoon, Hyeon Hwang, and Jaewoo Kang. Outlier-safe pre-training for robust 4-Bit quantization of large language models. *arXiv preprint arXiv:2506.19697*, 2025.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Artem Riabinin, Egor Shulgin, Kaja Gruntkowska, and Peter Richtárik. Gluon: Making Muon & Scion great again! (bridging theory and practice of LMO-based optimizers for LLMs). *arXiv preprint arXiv:2505.13416*, 2025.

Yichuan Shao, Shiqian Weng, Haijing Sun, Qian Gao, Le Zhang, Zhiqiang Mao, Shuai Xu, Zhitao Zhang, and Lei Xing. BDS-Adam optimizer integrating adaptive variance rectification with semi-adaptive gradient smoothing. *Scientific Reports*, 15(1):36906, 2025.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, 2018.

Shashank Subramanian, Robert M. Kirby, Michael W. Mahoney, and Amir Gholami. Adaptive self-supervision algorithms for physics-informed neural networks. *arXiv preprint arXiv:2207.04084*, 2022.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 2021.

Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2020.

Huizhuo Yuan, Yifeng Liu, Shuang Wu, Xun Zhou, and Quanquan Gu. MARS: Unleashing the power of variance reduction for training large models. In *International Conference on Machine Learning*, 2025.

A FUTURE WORK

- **Theory and guarantees.** Develop stability and convergence analyses for orthogonalized updates combined with bounded, damping-only feature-wise clipping. A promising direction is to formalize a Lyapunov-style descent argument under explicit alignment conditions and bounded-update properties, and to connect empirical spectral diagnostics (e.g., the effective update statistic $A_{\text{eff}} = W^\top \Delta W$) to provable stability regimes.
- **Generalizing feature axes beyond 2D matrices.** Extend EnergyCol-style clipping to convolutional kernels, embeddings, and higher-order tensors by defining principled "feature axes" (e.g., input-channel, attention-head, or group dimensions). We also plan to explore block-wise and low-rank variants that preserve interpretability while reducing per-step overhead.
- **Adaptive burst modeling with transparent control.** Replace fixed clipping hyperparameters (e.g., $c_{\min}$, α, update period K) with lightweight, interpretable adaptations driven by online tail statistics of the energy distribution (quantiles, kurtosis, robust outlier scores), while maintaining the damping-only constraint and avoiding hidden amplification.
- **Systems and numerical precision.** Improve the efficiency and robustness of Newton–Schulz orthogonalization in mixed precision and distributed settings. This includes precision-aware kernels, communication-efficient implementations, and amortized/approximate orthogonalization strategies that retain most of the directional benefit at lower cost.
- **Broader robustness regimes and downstream impact.** Evaluate TRASMUON under realistic forms of nonstationarity common in large-scale training (curriculum shifts, domain-mixture changes, sequence-length spikes, and data-quality transitions), and study how clipping statistics correlate with downstream robustness and generalization.

B TRASMUON ALGORITHM (EXTENDED)

TrasMuon explicitly decouples *update geometry* (direction) from *step-size control* (magnitude). For a matrix parameter $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ with stochastic gradient $G_t = \nabla_W \mathcal{L}(W_t)$, it applies multiplicative coupling as

$$\Delta W_t \;=\; -\,\hat{\eta}_t\, O_t^{\text{base}}\, \text{diag}(c_t), \tag{10}$$

where $O_t^{\text{base}} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ provides a structured direction, $\hat{\eta}_t$ is a row-wise RMS-calibrated global step size, and $c_t \in [c_{\min}, 1]^{d_{\text{in}}}$ is a *feature-axis* (column-wise) damping vector (damping-only, no amplification).

B.1 ORTHOGONALIZED DIRECTIONS VIA NEWTON–SCHULZ

TrasMuon maintains an exponential moving average of gradients

$$M_t \;=\; \beta_1 M_{t-1} + (1-\beta_1) G_t. \tag{11}$$

To extract a rotation-robust, near-isometric direction, we approximate the polar factor of M_t. For numerical stability of Newton–Schulz (NS) iterations, we remove the scale gauge by RMS-normalizing

$$\tilde{M}_t \;=\; \frac{M_t}{\|M_t\|_F / \sqrt{d_{\text{out}} d_{\text{in}}} + \epsilon}, \tag{12}$$

and apply T NS steps to obtain

$$O_t \;\approx\; \text{NS}(\tilde{M}_t; T) \;\approx\; \tilde{M}_t (\tilde{M}_t^\top \tilde{M}_t)^{-1/2}, \tag{13}$$

yielding a structured direction that is less sensitive to axis rotations than elementwise or diagonal preconditioning (e.g., Muon-style orthogonalized updates).

Algorithm 1 TRASMUON: Muon + Adaptive Scaling + Trust Region + Schedule-Free Smoothed

Input: $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, base lr η, $\beta_1, \beta_2, \epsilon$, NS steps T, weight decay λ, $c_{\min}, \alpha, \beta_E, \beta_c$, trigger k, update period K, warmup T_w, mix ρ
Initialize $M \leftarrow 0$, $v^{\text{row}} \leftarrow 0$, $E^{\text{ref}} \leftarrow 0$, $c^{\text{ema}} \leftarrow \mathbf{1}$, $c^{\text{last}} \leftarrow \mathbf{1}$, $S \leftarrow 0$, $C \leftarrow \mathbf{0}$, $\gamma_t \leftarrow \eta$
repeat
  $G \leftarrow \nabla_W \mathcal{L}(W)$; $W \leftarrow (1 - \eta\lambda)W$
  Momentum: $M \leftarrow \beta_1 M + (1 - \beta_1)G$
  Orthogonalized direction: $O \leftarrow \text{NS}(\tilde{M}; T)$
  $v^{\text{row}} \leftarrow \beta_2 v^{\text{row}} + (1 - \beta_2)\,\text{mean}_j(O_{\cdot j}^2)$
  $O^{\text{base}} \leftarrow \text{diag}\big((v^{\text{row}} + \epsilon)^{-1/2}\big)\, O$
  Calibration: $\hat{\eta} \leftarrow \eta \, \frac{\sqrt{d_{\text{out}} d_{\text{in}}}}{\|O^{\text{base}}\|_F + \epsilon}$
  Column energy: $E_j \leftarrow \sum_i M_{ij}^2$
  Robust reference: $E^{\text{cur}} \leftarrow \text{Quantile}_{0.5}(\{E_j\})$
  EMA smooth: $E^{\text{ref}} \leftarrow \beta_E E^{\text{ref}} + (1 - \beta_E)E^{\text{cur}}$
  Schedule-free accumulators: $S \leftarrow S + \gamma_t^2$, $C \leftarrow C + \gamma_t^2 c^{\text{last}}$
  $c^{\text{avg}} \leftarrow C/(S + \epsilon)$
  if $t > T_w$ **and** $t \bmod K = 0$ **then**
    $r_j \leftarrow E_j/(E^{\text{ref}} + \epsilon)$
    $c_j^{\text{clip}} \leftarrow \text{clip}\Big(\frac{1}{1+\alpha \log(1+r_j)},\, c_{\min},\, 1\Big)$
    Trigger (optional): $c_j^{\text{inst}} \leftarrow c_j^{\text{clip}}$ if $r_j > k$ else 1
    EMA smooth: $c^{\text{ema}} \leftarrow \beta_c c^{\text{ema}} + (1 - \beta_c)c^{\text{inst}}$
  end if
  Long-Short term Mixing: $c \leftarrow \mathbf{1}$ if $t \le T_w$ else $(1 - \rho)c^{\text{ema}} + \rho c^{\text{avg}}$
  $c^{\text{last}} \leftarrow c$ {cached between updates}
  Update: $W \leftarrow W - \hat{\eta}\,\big(O^{\text{base}} \odot \text{ExpandCols}(c)\big)$
until training ends

B.2 ROW-SECOND-MOMENT SCALING AND RMS-CALIBRATED STEP SIZE

Orthogonalization primarily shapes *direction*. To stabilize *magnitude* across layers and time, we apply lightweight row-wise second-moment scaling (as in NorMuon (Li et al., 2025)):

$$v_t^{\text{row}} \;=\; \beta_2 v_{t-1}^{\text{row}} + (1 - \beta_2)\,\text{mean}_j\big(O_{t,\cdot j}^{\odot 2}\big)\,, \tag{14}$$

$$O_t^{\text{base}} \;=\; \text{diag}\big((v_t^{\text{row}} + \epsilon)^{-1/2}\big)\, O_t. \tag{15}$$

Row scaling addresses row-wise heterogeneity, while a *global* row-wise calibration controls the update norm. We set

$$\hat{\eta}_t \;=\; \eta \cdot \frac{\sqrt{d_{\text{out}} d_{\text{in}}}}{\|O_t^{\text{base}}\|_F + \epsilon}, \tag{16}$$

so that the per-step RMS magnitude of ΔW_t is on the order of η. Since $c_t \le \mathbf{1}$ elementwise, equation 16 also implies an explicit Frobenius-norm bound $\|\Delta W_t\|_F \le \eta\sqrt{d_{\text{out}} d_{\text{in}}}$ (up to ϵ), reducing sensitivity to layer shape and fluctuations in the orthogonalized direction (Bernstein & Newhouse, 2025; Large et al., 2024).

B.3 ENERGY-BASED FEATURE-WISE TRUST-REGION CLIPPING

Motivation. In practice, instability often arises from *bursty magnitudes* that concentrate on a small subset of feature axes (columns), causing loss spikes and narrowing the stable learning-rate region. TrasMuon therefore introduces *feature-wise clipping*: it dampens only the high-energy feature directions while preserving the Muon-like direction structure in O_t^{base}.

Column energy and a robust reference. The column energy is measured from momentum M_t:

$$E_{t,j} \;=\; \sum_{i=1}^{d_{\text{out}}} M_{t,ij}^2, \qquad j = 1, \ldots, d_{\text{in}}. \tag{17}$$

We summarize the typical energy level at step t by a quantile statistic

$$E_t^{\mathrm{cur}} \;=\; \mathrm{Quantile}_q\big(\{E_{t,j}\}_{j=1}^{d_{\mathrm{in}}}\big), \qquad q = 0.5, \tag{18}$$

and maintain a running reference via an EMA updated every step:

$$E_t^{\mathrm{ref}} \;=\; \beta_E E_{t-1}^{\mathrm{ref}} + (1-\beta_E)\, E_t^{\mathrm{cur}}. \tag{19}$$

Using a quantile (median) yields a high-breakdown reference: the sample median has a 50% breakdown point, so a sparse set of high-energy columns cannot arbitrarily inflate E_t^{ref} and thereby "move the clipping threshold" in response to the outliers being clipped (Hampel et al., 1986; Huber, 1981).

Relative ratio and clipping-style damping. We define a dimensionless ratio as follow:

$$r_{t,j} \;=\; \frac{E_{t,j}}{E_t^{\mathrm{ref}} + \epsilon}. \tag{20}$$

A hard energy cap $E_{t,j} \le k\, E_t^{\mathrm{ref}}$ gives the column-wise analogue of norm clipping:

$$c_{t,j}^{\mathrm{hard}} \;=\; \min\left(1, \sqrt{\frac{k\, E_t^{\mathrm{ref}}}{E_{t,j} + \epsilon}}\right), \tag{21}$$

which enforces $r_{t,j} \le k$ after rescaling. TrasMuon leverages a smooth, numerically stable *soft clipping* rule:

$$c_{t,j}^{\mathrm{raw}} \;=\; \frac{1}{1 + \alpha \log(1 + r_{t,j})}, \qquad c_{t,j}^{\mathrm{gate}} \;=\; \mathrm{clip}\big(c_{t,j}^{\mathrm{raw}},\, c_{\min},\, 1\big)\,. \tag{22}$$

This rule is bounded and avoids power-law instabilities as $r_{t,j} \to 0$. Importantly, $c_{t,j} \le 1$ for all j, so the mechanism is strictly damping-only and can be interpreted as a trust-region safety mechanism in *feature space*.

Triggered vs. continuous clipping. Optionally, damping can be also applied only when $r_{t,j}$ exceeds a triggering threshold k:

$$\bar{c}_{t,j} \;=\; \begin{cases} c_{t,j}^{\mathrm{gate}}, & r_{t,j} > k, \\ 1, & \text{otherwise}, \end{cases} \tag{23}$$

so that non-burst columns remain unchanged and the gate acts as an event-driven clip.

B.4 Temporal Smoothing and Schedule-Free Averaging

To reduce short-term noise and avoid sensitivity to the gate-update period, we smooth the instantaneous clip by first applying EMA smoothing:

$$c_t^{\mathrm{ema}} \;=\; \beta_c c_{t-1}^{\mathrm{ema}} + (1-\beta_c) c_t^{\mathrm{inst}}, \tag{24}$$

where c_t^{inst} denote the clip applied at step t (either c_t^{gate} or $\bar{c}_t$ depending on triggering). Second, we maintain a schedule-free average using an effective step weight γ_t (default $\gamma_t = \eta$) (Defazio et al., 2024). Define the scalar accumulator $S_t \in \mathbb{R}$ and vector accumulator $C_t \in \mathbb{R}^{d_{\mathrm{in}}}$:

$$S_t \;=\; S_{t-1} + \gamma_t^2, \qquad C_t \;=\; C_{t-1} + \gamma_t^2 c_{t-1}^{\mathrm{last}}, \qquad c_t^{\mathrm{avg}} \;=\; \frac{C_t}{S_t + \epsilon}, \tag{25}$$

where c_{t-1}^{last} is the most recently applied clip (cached between updates). We then mix short- and long-term estimates:

$$c_t \;=\; (1-\rho) c_t^{\mathrm{ema}} + \rho c_t^{\mathrm{avg}}, \qquad c_t^{\mathrm{last}} \leftarrow c_t. \tag{26}$$

This effective-time averaging reduces sensitivity to warmup length and total training steps, and prevents bias when the raw clip is computed only every K steps.

B.5 Final TrasMuon update

Substituting the components into equation 10 yields the final update

$$\Delta W_t \;=\; -\hat{\eta}_t\, O_t^{\mathrm{base}}\, \mathrm{diag}(c_t), \tag{27}$$

which preserves Muon-like directional geometry via O_t^{base} while controlling bursty magnitudes along feature axes through damping-only feature clipping c_t.

C Convergence Analysis

Scope. We provide a convergence *framework* for TrasMuon/energy clipping updates, separating (i) unconditional algebraic properties (bounded update norm; damping-only contraction) from (ii) mild alignment assumptions connecting the structured update to descent.

Damping-only contraction. For any matrix A and any $c \in [0,1]^n$, right-multiplication by $\mathrm{diag}(c)$ cannot increase the Frobenius norm:

$$\|A\,\mathrm{diag}(c)\|_F \le \|A\|_F. \tag{28}$$

RMS calibration. With $\hat{\eta}_t = \eta\sqrt{mn}/(\|O_t^{\mathrm{base}}\|_F + \epsilon)$ and damping-only $c_t \le \mathbf{1}$, the update norm is uniformly bounded:

$$\|\Delta W_t\|_F \le \eta\sqrt{mn} \qquad \forall t, \tag{29}$$

independently of transient gradient spikes.

Stationarity under smoothness and alignment. Under standard L-smoothness and a mild alignment condition (Appendix C), TrasMuon satisfies an expected first-order stationarity bound of the form

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(W_t)\|_F^2 \;\le\; \frac{\mathbb{E}[f(W_0)] - f^\star}{\mu\,\eta\,T} \;+\; \frac{L}{2\mu}\,\eta\,mn, \tag{30}$$

for a constant $\mu > 0$ capturing effective descent. Importantly, the EMA/schedule-free construction of c_t only affects how the clip is computed, while the theory relies solely on the invariant $c_{t,j} \in [c_{\min}, 1]$.

C.1 Algorithmic abstraction

Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ and let G_t be a stochastic gradient at W_t. We study updates of the form

$$W_{t+1} \;=\; (1-\eta\lambda)\,W_t \;+\; \Delta W_t, \qquad \Delta W_t \;:=\; -\hat{\eta}_t U_t, \qquad U_t := O_t^{\mathrm{base}}\mathrm{diag}(c_t), \tag{31}$$

with RMS-calibrated step size

$$\hat{\eta}_t \;=\; \eta\cdot\frac{\sqrt{mn}}{\|O_t^{\mathrm{base}}\|_F + \epsilon}, \tag{32}$$

and damping-only clipping

$$0 < c_{\min} \le c_{t,j} \le 1 \quad \forall j, t. \tag{33}$$

C.2 Deterministic algebraic properties

Lemma C.1 (Damping-only contraction)**.** *For any $A \in \mathbb{R}^{m\times n}$ and any $c \in [0,1]^n$, $\|A\,\mathrm{diag}(c)\|_F \le \|A\|_F$.*

Proof. $\|A\,\mathrm{diag}(c)\|_F^2 = \sum_{j=1}^n c_j^2\|A_{\cdot j}\|_2^2 \le \sum_{j=1}^n \|A_{\cdot j}\|_2^2 = \|A\|_F^2$. □

Lemma C.2 (Row-wise RMS calibration)**.** *Under equation 31–equation 33, for all t,*

$$\|\Delta W_t\|_F \le \eta\sqrt{mn}. \tag{34}$$

Proof. By Lemma C.1, $\|U_t\|_F = \|O_t^{\mathrm{base}}\mathrm{diag}(c_t)\|_F \le \|O_t^{\mathrm{base}}\|_F$. Thus

$$\|\Delta W_t\|_F = \hat{\eta}_t\|U_t\|_F \le \eta\sqrt{mn}\cdot\frac{\|O_t^{\mathrm{base}}\|_F}{\|O_t^{\mathrm{base}}\|_F + \epsilon} \le \eta\sqrt{mn}.$$

□

C.3 Assumptions for descent

Assumption C.3 (L-smoothness)**.** f is L-smooth with respect to Frobenius norm: $\|\nabla f(X) - \nabla f(Y)\|_F \le L\|X - Y\|_F$.

Assumption C.4 (Stochastic gradients)**.** $\mathbb{E}[G_t \mid W_t] = \nabla f(W_t)$ and $\mathbb{E}[\|G_t - \nabla f(W_t)\|_F^2 \mid W_t] \le \sigma^2$.

Assumption C.5 (Alignment on the realized update)**.** There exists $\mu_\Delta > 0$ such that for all t,

$$\mathbb{E}\big[\langle \nabla f(W_t), \Delta W_t\rangle \mid W_t\big] \;\le\; -\mu_\Delta\, \eta\, \|\nabla f(W_t)\|_F^2, \tag{35}$$

where $\Delta W_t = -\hat{\eta}_t U_t$.

C.4 Expected stationarity

Lemma C.6 (Smoothness descent)**.** *Under Assumption C.3, for any* Δ*,*

$$f(W_t + \Delta) \le f(W_t) + \langle \nabla f(W_t), \Delta\rangle + \frac{L}{2}\|\Delta\|_F^2. \tag{36}$$

Theorem C.7 (Expected stationarity for RMS-calibrated, damping-only updates)**.** *Assume C.3 and C.5 with* $\lambda = 0$*. Then for any* $T \ge 1$*,*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(W_t)\|_F^2 \;\le\; \frac{\mathbb{E}[f(W_0)] - f^\star}{\mu_\Delta\, \eta\, T} \;+\; \frac{L}{2\mu_\Delta}\, \eta\, mn, \tag{37}$$

where $f^\star = \inf_W f(W)$.

Proof. Apply Lemma C.6 with $\Delta = \Delta W_t$, take conditional expectation, use Assumption C.5, and bound $\|\Delta W_t\|_F^2$ by Lemma C.2. Sum over $t = 0, \dots, T-1$ and telescope $f(W_t)$. □

C.5 Extensions: weight decay, stochasticity, and PL

Weight decay. Decoupled weight decay can be handled by analyzing $f_\lambda(W) = f(W) + \frac{\lambda}{2}\|W\|_F^2$ or treating $(1 - \eta\lambda)W_t$ as an additional contraction term.

Stochastic gradients. Under Assumption C.4, the bound acquires an additional $\mathcal{O}(\eta\sigma^2)$ term as in standard SGD analyses.

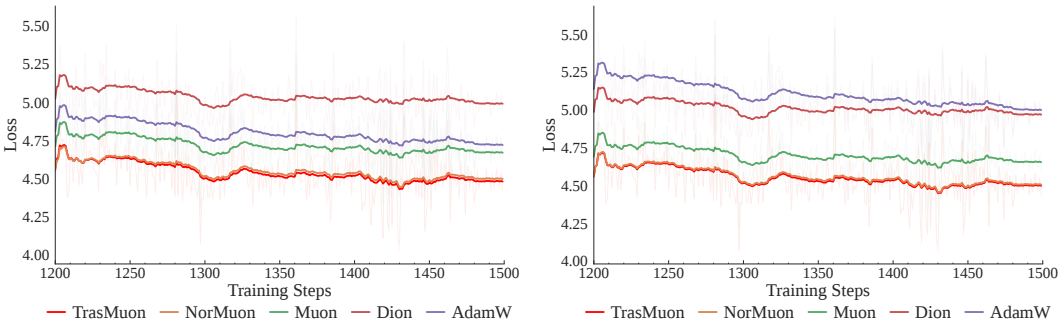
PL / strong convexity. If f satisfies the PL condition, one obtains linear-type convergence to an $\mathcal{O}(\eta mn)$ (or $\mathcal{O}(\eta\sigma^2)$) neighborhood.

D Supplementary Results: Language-Model Pretraining

Scope. This appendix provides supplementary visualizations for language-model pretraining and a minimal replication check. The intent is *not* to introduce new claims beyond the main text, but to (i) document late-stage behavior under the fixed training budget and (ii) reduce the risk that qualitative trends are specific to a single random seed.

D.1 Experiment Settings

Fixed-budget protocol and reporting (language model pretraining). TrasMuon is evaluated in a controlled, short-run pretraining-style setting and compared against four baseline optimizers: AdamW (Loshchilov & Hutter, 2019), Muon (Jordan et al., 2024), Dion (Ahn et al., 2025), and NorMuon (Li et al., 2025). Decoder-only Transformer models are trained from random initialization, including GPT-2 (Radford et al., 2019) and Qwen3-0.6B (Yang et al., 2025), on FineWeb-Edu (Lozhkov et al., 2024). All runs follow the same fixed-budget protocol as Section 4: training proceeds for 1500 optimization steps with sequence length 1024 and a fixed global (effective) batch size of 1024, corresponding to $1500 \times 1024 \times 1024 \approx 1.57 \times 10^9$ training tokens. Unless



(a) Warmup-enabled.

(b) Warmup-free.

Figure 2: **Qwen3-0.6B late-stage window (steps 1200–1500) under a fixed-budget protocol.** Training loss under warmup-enabled vs. warmup-free variants of the same warmup–stable–decay schedule with a final decay phase near the end of training.

otherwise stated, all optimizers share the same learning rate $\eta = 3.6 \times 10^{-3}$ and weight decay $\lambda = 5 \times 10^{-3}$, with no learning-rate sweep and no optimizer-specific retuning in this appendix. All runs follow the same default seeding behavior of the training stack. For NorMuon, the most influential optimizer-specific hyperparameter is the RMS normalization, which controls scale calibration of the orthogonalized update.

A warmup–stable–decay learning-rate schedule is adopted, and results are reported for two schedule variants: (i) a warmup-enabled configuration with 10% warmup and a final 20% decay phase, and (ii) the same schedule without warmup. All other training components, including the data pipeline, batching, tokenization, model architecture, and compute budget, are kept identical across optimizers. This section is treated as a fixed-hyperparameter pilot, where broader sweeps and multi-seed evaluations are deferred to future work.

Metrics. Training loss is reported as a high-resolution indicator of early-stage optimization dynamics and stability under an identical training budget. This appendix presents the corresponding loss trajectories to complement the main-text summaries.

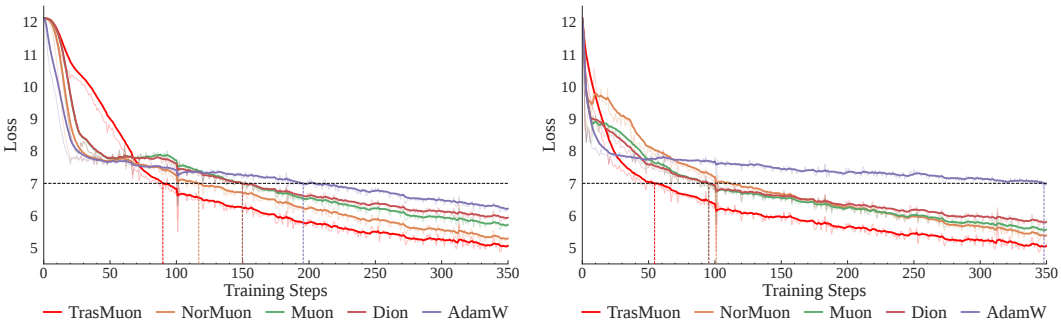
D.2 Qwen3-0.6B: Late-stage loss under a fixed 1500-step budget

The late-stage training window (steps 1200–1500) is examined to complement the early-stage analysis in the main text. Figure 2 shows the corresponding loss trajectories in this window for both schedule variants. These curves document late-stage behavior under a fixed training budget and are not intended to imply full convergence.

D.3 Qwen3-0.6B: Minimal replication check

The main text reports Qwen3-0.6B results under a single fixed seed. To reduce the possibility that the observed qualitative trends are specific to that particular random draw, the same Qwen3-0.6B runs are repeated with an additional random seed while keeping *all* other settings identical, including the data pipeline, token budget, model architecture, learning rate, weight decay, and schedule variant. These replication runs are not used for hyperparameter tuning and are included solely as a robustness check under an identical protocol.

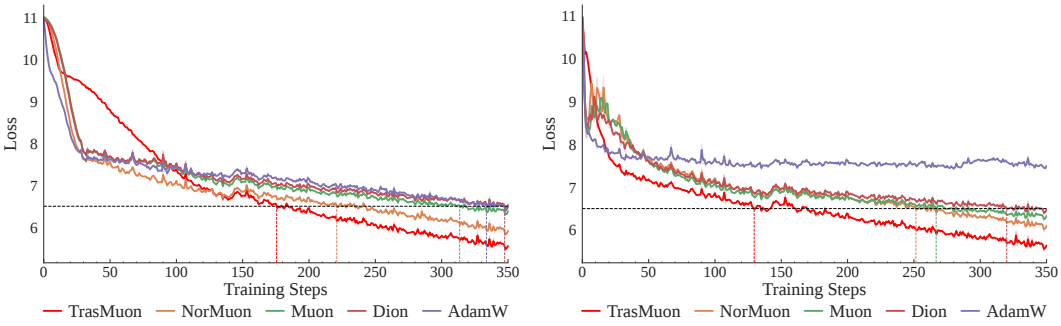
Figure 3 presents early-stage loss trajectories over steps 0–350 for the additional seed. Across both schedule variants, the qualitative optimizer behavior remains consistent with the main-text observations. No claim of statistical significance is made from this minimal replication; the results are provided as supplementary evidence of qualitative robustness under matched conditions.



(a) Warmup-enabled, seed 42.

(b) Warmup-free, seed 42.

Figure 3: **Qwen3-0.6B replication under an identical protocol (additional seed).** Early-stage training loss (steps 0–350) for an additional random seed, under the same configuration as the main-text experiment.



(a) Warmup-enabled.

(b) Warmup-free.

Figure 4: **GPT-2 Small early-stage dynamics (steps 0–350) under a fixed-budget protocol.** Warmup-enabled vs. warmup-free variants of the same warmup–stable–decay schedule.

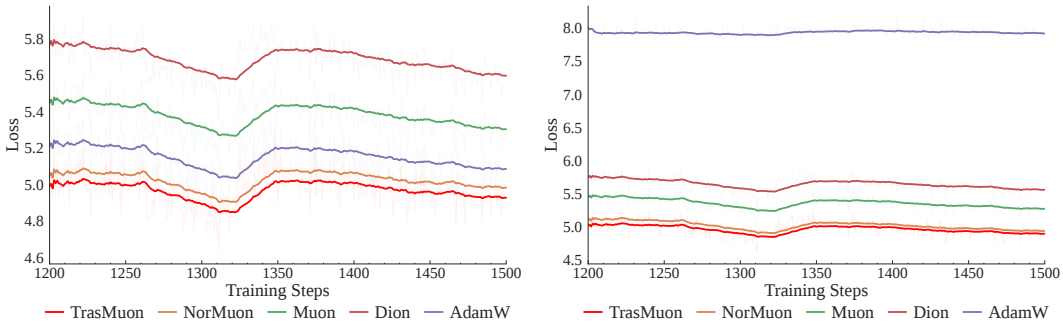
D.4 GPT-2 Small: Additional architecture under the same protocol

GPT-2 Small is additionally evaluated under the same fixed-budget protocol to assess whether the observed qualitative dynamics transfer to a different architecture. Figure 4 presents early-stage loss trajectories over steps 0–350 for both schedule variants, and Figure 5 reports the late-stage window over steps 1200–1500. These plots are included to complement the main-text results, and the late-stage window is reported for completeness rather than as evidence of full convergence.

E Vision Transformer Experiments

E.1 ImageNet-100 Data Source and Construction

To reduce dataset preparation overhead, we use a publicly available ImageNet-100 *image archive* hosted on Kaggle (Ambityga, 2021) as a convenient storage source. Importantly, the Kaggle archive is used *only* as a source of image files. Class membership and the train/validation split are defined strictly according to the ILSVRC-2012 (ImageNet-1k) specification. Concretely, a fixed subset of 100 ILSVRC-2012 classes is selected (specified by synset IDs) and retained images whose labels match these synsets; we then follow the standard ILSVRC-2012 train/validation split. As a sanity check, we verify (in code) the synset-to-index mapping, per-class image counts, and that no images outside the selected synsets are included. This ensures that the benchmark corresponds to a well-defined subset of ImageNet-1k, independent of the hosting platform.



(a) Warmup-enabled.

(b) Warmup-free.

Figure 5: **GPT-2 Small late-stage window (steps 1200–1500) under a fixed-budget protocol.** Training loss for warmup-enabled vs. warmup-free schedule variants.

Table 1: **ViT on ImageNet-100.** Final *validation* top-1 accuracy (mean ± standard deviation) over three random seeds (42, 43, 44).

Optimizer	Accuracy Mean	Accuracy Std
TrasMuon	**77.47%**	**0.34%**
NorMuon	77.10%	0.21%
Muon	69.69%	0.08%
AdamW	42.53%	4.38%

E.2 ImageNet-100 Experimental Protocol and Seed Allocation

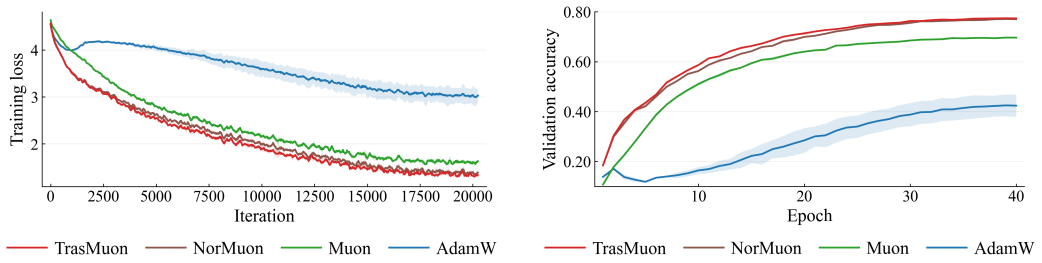
We evaluate optimization methods on ImageNet-100 using a ViT-Base/16 architecture at 224×224 resolution. Training follows a standard ViT/DeiT recipe (Dosovitskiy et al., 2021; Touvron et al., 2021), including random resized cropping, horizontal flipping, color jitter, RandAugment, and random erasing, together with label smoothing and Mixup (CutMix disabled).

Weight decay is applied with parameter grouping: LayerNorm and bias parameters use zero weight decay, and all remaining parameters use a fixed decay rate. Unless otherwise stated, all experiments use a base learning rate of 1×10^{-3} and a weight decay of 5×10^{-2}. We compare AdamW (Loshchilov & Hutter, 2019), Muon (Jordan et al., 2024), NorMuon (Li et al., 2025), and TrasMuon under the same model architecture, data pipeline, training schedule, and compute budget, using three random seeds (42, 43, 44). Optimizer-specific parameters follow the respective published defaults and our unified implementation. For each method, we report the mean and standard deviation of *validation* top-1 accuracy across seeds (Table 1).

E.3 ImageNet-100: Vision Transformer Training.

We evaluate the benefits of TrasMuon in a large-scale vision setting by training ViT-Base (Dosovitskiy et al., 2021) on ImageNet-100 due to limited computational resources, a 100-class subset of ImageNet-1k (ILSVRC-2012) (Deng et al., 2009), followed the standard ILSVRC-2012 train/validation protocol. Dataset construction, class specification, and implementation details for ViT training are provided in Appendix E.1 and E.2. We compared AdamW, Muon, NorMuon, and TrasMuon, using identical training budgets and hyperparameters. Results show training loss and *validation* top-1 accuracy, aggregated over three random seeds.

Results. Across all optimizers evaluated with multi-seed runs, Muon, Normuon, and TrasMuon consistently improve optimization behavior and validation accuracy over AdamW in these experiments, demonstrating the advantage of optimizers, which update based on structured, near-orthogonal update directions. illustrated in Fig. 6a and Fig. 6b, TrasMuon achieves the fastest loss reduction, the highest validation accuracy, and reduced variability across different seeds and vari-



(a) Training loss vs. iteration.

(b) Validation top-1 accuracy vs. epoch.

Figure 6: ViT-Base training on ImageNet-100. Multi-seed results (mean ± std over three seeds: 42, 43, 44) for 4 optimizers. Shaded regions denote variability across seeds.

Table 2: **ViT on CIFAR-100 with column-localized gradient bursts.** Final top-1 test accuracy (mean ± standard deviation) over three seeds (42, 43, 44).

Optimizer	Accuracy Mean	Accuracy Std
TrasMuon	**58.77%**	**0.22%**
NorMuon	58.31%	0.52%
Muon	57.48%	0.52%
AdamW	35.03%	5.05%

ous compared optimizers. Moreover, CIFAR-100 has been evaluated robustness under controlled column-localized burst injection in Appendix E.4.

E.4 CIFAR-100: Column-Energy Stress Test

Setup (stress-test benchmark). We conduct a controlled stress test on CIFAR-100 using a Vision Transformer to assess optimizer robustness under axis-localized nonstationarity. Unless otherwise specified, all runs use 30 epochs, batch size 128, learning rate 1×10^{-3}, weight decay 5×10^{-3}, and identical data loading and preprocessing. We report mean and standard deviation of test accuracy over three random seeds (42, 43, 44).

Burst injection. To introduce structured nonstationarity without changing the data distribution, we inject sparse *column-localized gradient bursts* into selected large 2D parameter matrices (attention and MLP projections). The goal of this protocol is not to claim that injected bursts match the exact distribution of naturally occurring outliers, but to provide a reproducible mechanism-level stressor that concentrates energy along a small subset of feature axes. Crucially, the same burst pattern (targeted layers, selected columns, and timesteps) is applied across optimizers by fixing the burst random seed, enabling direct, fair comparison of optimizer responses. Full implementation details are provided in Appendix E.5.

Observed behavior. Table 2 summarizes test accuracy under burst injection. Across this stress setting, Muon-family optimizers maintain higher accuracy and lower variability than AdamW. Normalization-based variants reduce variance relative to Muon, while TrasMuon attains the highest mean accuracy and the smallest spread across seeds in this configuration.

E.5 Burst Injection Protocol in CIFAR-100 Benchmark

We define a column-wise gradient burst operator applied to selected 2D weight matrices to induce controlled column-energy spikes without altering the data distribution. At each burst step, we select k column indices (random or fixed, as specified) and perturb each selected column by adding a normalized random direction:

$$g_{:,j} \leftarrow g_{:,j} + \alpha \cdot \frac{u}{\|u\|_2 + \epsilon}, \quad u \sim \mathcal{N}(0, I).$$

The amplitude α can be specified as a fixed absolute value or scaled relative to the current gradient magnitude using a Frobenius-normalized reference:

$$\alpha = \rho \cdot \frac{\|g\|_F}{\sqrt{d_{\text{out}} d_{\text{in}}}},$$

optionally clipped by a maximum threshold. Bursts occur every T optimization steps after an optional warmup phase and target only designated 2D layers (e.g., attention projections and MLP weights). Burst events and optimizer internal statistics (including feature-wise clipping coefficients, when applicable) are logged at the same timesteps to enable direct alignment between perturbations and optimizer responses.

F PINNs With Random-ROI Sampling Stress Test

Adaptive collocation in physics-informed neural networks (PINNs) is often necessary to resolve localized errors and stiff PDE behavior, where uniform sampling under-resolves difficult regions (Gao et al., 2023; Subramanian et al., 2022; Wu et al., 2023). Here we use region-of-interest (ROI) densification as a *controlled nonstationarity* mechanism to stress-test optimizer robustness: periodically concentrating interior collocation points in a small subregion induces distribution shifts in the residual samples, perturbing gradient statistics in a reproducible way.

Helmholtz equation setup. On $\Omega = [0,1]^2$, we consider

$$\Delta u(x) + \kappa^2 u(x) = f(x), \qquad u|_{\partial\Omega} = 0, \tag{38}$$

with the manufactured solution

$$u^\star(x,y) = \sin(\pi k x)\sin(\pi k y), \qquad \kappa = \pi k. \tag{39}$$

which yields $f(x) = -(\pi k)^2 u^\star(x)$.

We train an MLP $u_\theta : \Omega \to \mathbb{R}$ by minimizing

$$\mathcal{L}(\theta) = \mathbb{E}\left[\tfrac{1}{2} r_\theta(x)^2\right] \; + \; \lambda_b \, \mathbb{E}\left[\tfrac{1}{2}\big(u_\theta(x) - u^\star(x)\big)^2\right] \tag{40}$$

$$r_\theta(x) = \Delta u_\theta(x) + \kappa^2 u_\theta(x) - f(x). \tag{41}$$

and report the relative error on a fixed evaluation grid

$$\text{rel-}L_2(u_\theta, u^\star) = \frac{\|u_\theta - u^\star\|_2}{\|u^\star\|_2}. \tag{42}$$

ROI sampling protocol. To emulate adaptive densification, we impose a controlled, time-varying interior sampling distribution. We consider the Helmholtz equation on $\Omega = [0,1]^2$ with homogeneous Dirichlet boundary conditions and a manufactured solution $u^\star$. We run 4000 optimization steps and introduce nonstationary ROI events after step $t_0 = 1000$. At each step we sample $N_r = 1024$ interior points and $N_b = 256$ boundary points, with boundary weight $\lambda_b = 100$. We evaluate every 200 steps on a fixed 128×128 grid.

Nonstationary ROI events (distribution shift). Starting from step $t_0 = 1000$, we trigger ROI events every $K_{\text{out}} = 20$ steps. At an ROI event step t, interior points are sampled from a mixture distribution

$$p_t(x) \;=\; (1-\alpha)\, p_0(x) + \alpha\, p_{\text{roi}}^{(t)}(x), \qquad \alpha = 0.95, \tag{43}$$

where p_0 is uniform over Ω and $p_{\text{roi}}^{(t)}$ is uniform over the selected ROI patch $\Omega_{\text{roi}}^{(t)} = [x_0, x_1] \times [y_0, y_1]$. This yields repeated, time-varying distribution shifts that mimic practical ROI/adaptive refinement policies in PINNs.

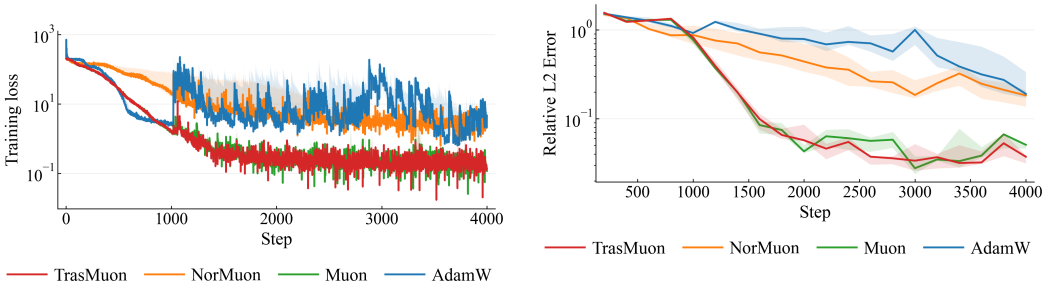
Random ROI patch pool (reproducible). To avoid sensitivity to a single ROI location, $\Omega_{\text{roi}}^{(t)}$ is chosen from a fixed pool (Table 3) using a deterministic `step_hash` seeding rule. Thus ROI locations vary across events while remaining fully reproducible given the experiment configuration and the training step index.

Table 3: ROI patch pool used for ROI events (rectangles are $[x_0, x_1] \times [y_0, y_1]$).

ROI patches
Corners: $[0.00, 0.03] \times [0.00, 0.03]$, $[0.97, 1.00] \times [0.00, 0.03]$, $[0.00, 0.03] \times [0.97, 1.00]$, $[0.97, 1.00] \times [0.97, 1.00]$
Edges: $[0.48, 0.53] \times [0.00, 0.05]$, $[0.48, 0.53] \times [0.95, 1.00]$, $[0.00, 0.05] \times [0.48, 0.53]$, $[0.95, 1.00] \times [0.48, 0.53]$
Interior: $[0.20, 0.25] \times [0.20, 0.25]$, $[0.45, 0.50] \times [0.10, 0.15]$, $[0.10, 0.15] \times [0.55, 0.60]$, $[0.60, 0.65] \times [0.60, 0.65]$

ROI-local evaluation. To quantify localized disturbance and recovery, at ROI event steps we additionally compute an ROI-local relL_2 on a 64×64 grid restricted to $\Omega_{\text{roi}}^{(t)}$. We estimate non-ROI error by sampling 4096 points from $\Omega \setminus \Omega_{\text{roi}}^{(t)}$. These diagnostics separate global convergence from localized behavior under distribution shifts.

F.1 PINNs Benchmark: ROI Sampling as a Nonstationary Stress Test



(a) training objective (estimated on the time-varying sampling distribution p_t)

(b) domain-wide relative L_2 error evaluated on a fixed grid. ROI events start at step 1000 and repeat every 20 steps

Figure 7: **PINN Helmholtz (k=2) under random ROI sampling shifts.** Curves show the mean over seeds, and shaded regions indicate variability across seeds.

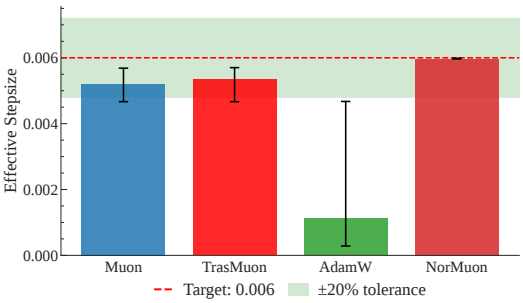
PINN ROI-sampling stress test: convergence and robustness. Figure 7a and 7b compare Muon and TrasMuon on Helmholtz (k=2) under a controlled nonstationary ROI-sampling protocol, where ROI events start at step 1000 and recur every 20 steps. During the initial stationary phase (before ROI events), both methods exhibit nearly identical optimization trajectories in terms of training loss and domain-wide relative L_2 error, indicating that TrasMuon does not incur a measurable overhead or degradation under standard uniform sampling.

After ROI events begin, the training objective becomes significantly more variable due to the induced distribution shifts in interior collocation points. In this nonstationary regime, TrasMuon maintains comparable or slightly lower training loss while exhibiting reduced extreme fluctuations, consistent with its design goal of suppressing bursty, feature-localized updates. These results support the conclusion that TrasMuon preserves baseline convergence under stationary sampling, while improving stability and final solution accuracy under controlled, nonstationary ROI sampling shifts.

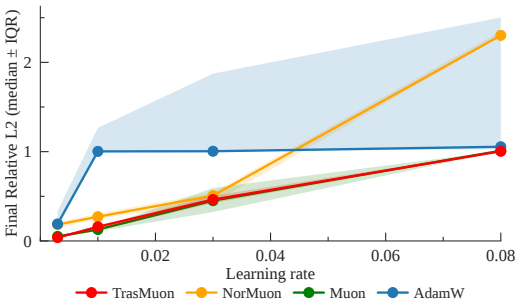
F.2 Step-Size Alignment and Learning-Rate Sensitivity (PINNs Helmholtz, $k = 2$)

Motivation. Nominal learning rates are not directly comparable across update rules because different optimizers can induce different *effective* parameter-space step magnitudes. To reduce the confound that performance differences are driven by trivial step-size mismatches, we complement the main comparison with (i) a short step-size alignment diagnostic and (ii) a shared learning-rate (LR) sweep.

Step-size alignment diagnostic. Figure 8a reports the achieved effective step size, computed from parameter differences during an initial stationary window (before any ROI perturbations are intro-



(a) **Achieved effective step size** during the stationary alignment window (before ROI events). The dashed line is the target, and the shaded region indicates tolerance. Error bars denote variability across seeds.



(b) **LR sensitivity** of final relative L_2 error under a shared LR sweep. Lines show the median across seeds and shaded bands indicate the IQR.

Figure 8: **PINNs Helmholtz ($k = 2$): step-size alignment and LR sensitivity.** (a) Effective step-size alignment reduces trivial magnitude confounds when comparing optimizers with different update rules. (b) A shared LR sweep summarizes sensitivity to step-size calibration via final relative L_2 error.

duced),

$$s_t = \frac{\|\Delta\theta_t\|_2}{\sqrt{P}}, \qquad P = \dim(\theta).$$

We target a fixed reference magnitude (dashed line) with a tolerance band (shaded region). Muon and TRASMUON attain comparable achieved step sizes within the tolerance range, supporting that subsequent robustness comparisons are not explained by a simple global step-size discrepancy.

Learning-rate sweep. Figure 8b shows the final relative L_2 error under a shared LR sweep. Shaded bands summarize variability across random seeds (median with an interquartile range). Both methods exhibit the expected degradation as LR increases beyond the stable region. Together with the alignment diagnostic, this sweep provides a complementary view of optimizer sensitivity to step-size calibration under the same training and sampling protocol.

F.3 PINNs Diagnostics: Metric Distributions Across Seeds

We visualize the distribution of key robustness and accuracy metrics across random seeds for the PINN Helmholtz benchmark($k = 2$). This figure serves as a distributional check to ensure that the reported trends are not driven by a single favorable run.

G Controlled Diagnostics

This appendix provides protocol-level details and supporting evidence for the controlled diagnostics study. The intent is two-fold: (i) to make the stress protocol fully reproducible, and (ii) to document a minimal, time-aligned evidence chain that is *consistent with* the intended energy-indexed, feature-wise clipping mechanism under a controlled intervention. We do not introduce new claims beyond Section G.1.

G.1 Controlled diagnostics: Column-Localized Outliers and Energy-Based Feature Clipping

We design a controlled toy problem to validate the *feature-wise clipping* mechanism in TRASMUON under intermittent, column-localized bursts. We optimize a matrix parameter $W \in \mathbb{R}^{d\times d}$ under the quadratic objective

$$\min_W \; f(W) \;=\; \tfrac{1}{2}\|AWB - T\|_F^2, \tag{44}$$

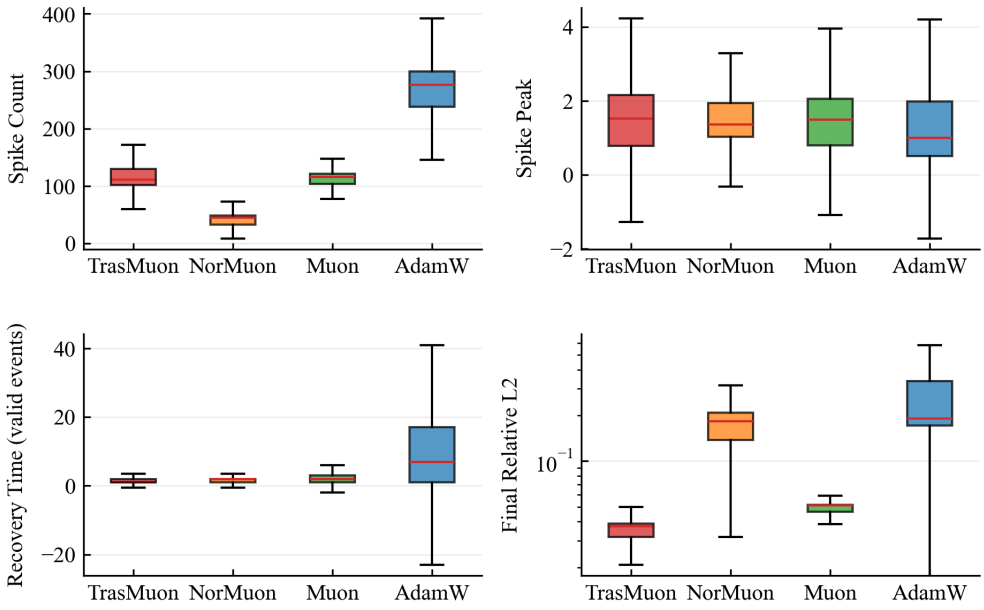


Figure 9: **PINNs Helmholtz($k = 2$): metric distributions across seeds.** Boxplots summarize spike count, spike peak, recovery time (valid events), valid-event rate, and final relative L_2 error across seeds under the same training protocol. Each box shows the median and interquartile range (IQR); whiskers indicate the remaining spread.

where $A = U\Sigma_A U^\top$ and $B = V\Sigma_B V^\top$ with random orthogonal U, V, and diagonal spectra Σ_A, Σ_B chosen to yield a target condition number $\kappa \in \{10^2, 10^4, 10^6\}$. This construction controls stiffness while allowing us to randomize nuisance rotations across runs.

Column-localized outlier injection (stress protocol). To emulate rare, feature-localized gradient domination, every K_{out} steps we inject an outlier event that amplifies a small subset of columns in a fixed feature basis. Concretely, for momentum M_t we select a set $\mathcal{J}$ of $s \ll d$ column indices and apply a multiplicative burst

$$\widetilde{M}_{t,\cdot j} = \begin{cases} a\, M_{t,\cdot j}, & \text{if } j \in \mathcal{J}, \\ M_{t,\cdot j}, & \text{otherwise}, \end{cases} \tag{45}$$

with burst amplitude $a > 1$. This perturbation produces abrupt increases in column energy $E_{t,j} = \sum_i \widetilde{M}^2_{t,ij}$ while leaving the underlying objective equation 44 unchanged.

Preserving feature semantics. Because TRASMUON's clipping is axis-aligned (column-wise), the stress protocol is evaluated under a `fix_V=True` setting, i.e., the column basis is preserved across training and across injected events. We additionally report a boundary condition where the column basis is randomized (`fix_V=False`); in that case, injected energy disperses across columns and feature-wise clipping is not expected to yield an advantage.

Metrics. We track (i) spike count and (ii) final objective value, reporting median and IQR over multiple seeds/rotations.

Closed-loop response. Figure 10 shows that TRASMUON reduces burst-induced loss spikes and improves convergence relative to the NorMuon backbone under matched compute. Figure 11 provides mechanism-level evidence consistent with a closed-loop response: outlier events increase the relative column-energy ratio (e.g., $r_{q95}/r_{\max}$), which is immediately followed by a decrease in the *applied* clipping signal (tracked by $c_{\text{used,min}}$), thereby damping the burst and suppressing spikes.

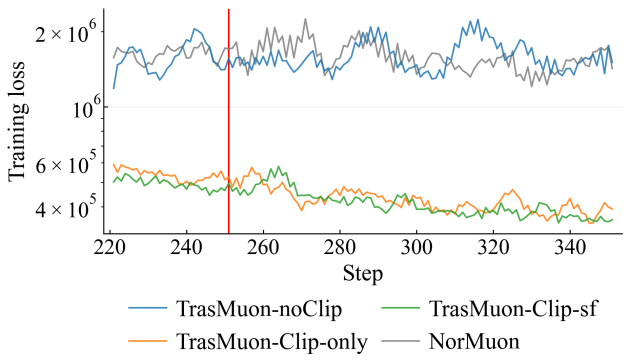


Figure 10: **Column outlier injection.** Loss trajectories in a window around an outlier event. Vertical markers indicate outlier steps.

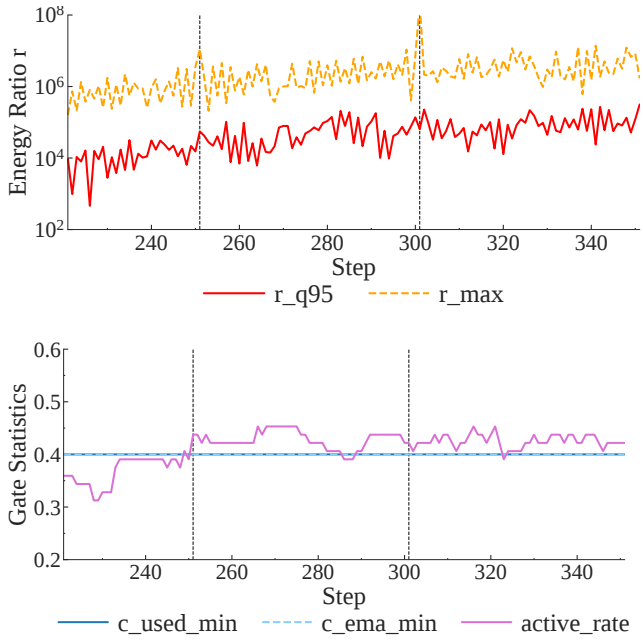


Figure 11: **Closed-loop clipping evidence.** Outlier events increase the column-energy ratio in log-scale (top), followed by stronger *feature-wise clipping* in the applied coefficients (bottom; $c_{\text{used,min}}$).

Not a trivial step-size reduction. To rule out the confound that improvements arise from a global effective step-size change, we include a TRASMUON-NOCLIP ablation that disables feature-wise clipping while keeping all other components identical. As summarized in Table 4, TRASMUON-NOCLIP behaves similarly to NorMuon, whereas enabling clipping yields a clear reduction in spike statistics and a large improvement in the final objective, isolating the contribution of feature-wise clipping.

Boundary condition (feature semantics broken). When the column basis is randomized, the advantage of feature-wise clipping diminishes, consistent with the intended mechanism: axis-aligned clipping requires a meaningful feature basis.

G.2 MINIMAL EVIDENCE CHAIN (TIME-ALIGNED OBSERVABLES)

Section G.1 argues that spike suppression is consistent with the following ordering under a controlled intervention: *outlier injection* $\rightarrow$ *relative energy increases* $\rightarrow$ *stronger applied clipping* $\rightarrow$ *reduced*

Table 4: **Controlled diagnoise summary under `fix_V=True`.** TRASMUON-NOCLIP removes feature-wise clipping while keeping the rest identical. We report median with IQR across runs; lower is better.

Method	Spike Count	Final Loss
NorMuon	44 (35,56)	1.3e+06 (1.0e+06,1.6e+06)
TRASMUON-noClip	48 (38,56)	1.1e+06 (8.9e+05,1.9e+06)
TRASMUON-Clip-only	**28** (24,34)	2.4e+05 (2.0e+05,2.8e+05)
TRASMUON-Clip-sf	30 (24,36)	**2.0e+05** (1.6e+05,2.7e+05)

loss spikes. We summarize the corresponding observables, which are directly logged and visualized in Fig. 11.

(1) Outlier injection increases relative energy. Eq. equation 45 increases the column energies $E_{t,j} = \sum_i \widetilde{M}^2_{t,ij}$ for $j \in \mathcal{J}$. We track the relative energy ratio

$$r_{t,j} \;=\; \frac{E_{t,j}}{E_t^{\mathrm{ref}} + \epsilon}, \tag{46}$$

where E_t^{ref} is the running reference used by TRASMUON. We visualize robust summaries such as r_{q95} and $r_{\max}$, which rise at injected outlier steps (Fig. 11, top).

(2) Higher relative energy is followed by stronger applied clipping. TRASMUON produces damping-only clipping coefficients $c_{t,j}^{\mathrm{used}} \in [c_{\min}, 1]$ that decrease with $r_{t,j}$. Consistent with this design, outlier steps are followed by a decrease in the applied signal, visible via $c_{\mathrm{used,min}} = \min_j c_{t,j}^{\mathrm{used}}$ (Fig. 11, bottom). This time alignment is consistent with the intended ordering "energy rise $\rightarrow$ clipping increase" under the intervention.

(3) Applied clipping attenuates column updates (selectively). Given the matrix-form update,

$$\Delta W_t \;=\; -\,\hat{\eta}_t\, O_t^{\mathrm{base}}\, \mathrm{diag}(c_t^{\mathrm{used}}), \tag{47}$$

each column update magnitude is scaled by $c_{t,j}^{\mathrm{used}}$:

$$\|\Delta W_{t,\cdot j}\|_2 \;=\; c_{t,j}^{\mathrm{used}}\; \hat{\eta}_t\; \|O_{t,\cdot j}^{\mathrm{base}}\|_2 \;\le\; \hat{\eta}_t\; \|O_{t,\cdot j}^{\mathrm{base}}\|_2. \tag{48}$$

Thus clipping is *selective*: only columns with $c_{t,j}^{\mathrm{used}} < 1$ are damped, while the structured direction O_t^{base} is preserved.

(4) Spike suppression and objective improvement. Consistent with (1)–(3), TRASMUON reduces loss spikes around outlier events (Fig. 10) and achieves lower final objective values than the backbone under matched compute (Table 4). Spike metrics (count/peak) are computed using the same deterministic detection rule across methods; details are provided in our experiment scripts and plotting utilities.

Table 5: **Boundary condition effects(`fix_V=False`).** When feature/column semantics are broken by column-space mixing, the advantage of feature-wise clipping diminishes.

Method	Spike Count	Final Loss
NorMuon	79 (74,86)	**1.1e+06** (8.4e+05,1.6e+06)
TRASMUON-noClip	80 (74,87)	1.4e+06 (9.8e+05,1.7e+06)
TRASMUON-clip-only	74 (67,80)	1.5e+06 (1.2e+06,1.8e+06)
TRASMUON-clip+SF	**72** (65,80)	1.3e+06 (9.4e+05,1.7e+06)

Controls that break the chain. We include two controls that remove key requirements of the mechanism: (i) TRASMUON-NOCLIP sets $c_t^{\mathrm{used}} \equiv \mathbf{1}$, removing the attenuation in Eq. equation 48; empirically it behaves similarly to NorMuon (Table 4). (ii) Under `fix_V=False`, the injected energy is dispersed across columns, so clipping is no longer aligned with injected directions; correspondingly, the advantage of feature-wise clipping diminishes (Table 5).