AF-MUON: AN ADAMW-FREE MUON OPTIMIZER FOR TIED-EMBEDDING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Muon improves large-scale training by applying a spectral-norm steepest-descent update to matrix parameters, but practical models also contain parameter blocks that do not fit dense-matrix geometry. One important case is the tied vocabulary table, which appears in language models and other token generators and can receive multiple structurally different gradient sources, from sparse input lookups to dense output-classifier updates. In the reference recipe these blocks are handed to an auxiliary AdamW optimizer, which restores second-moment state and updates the aliased table as a generic tensor. We propose AF-Muon, an AdamW-free extension of Muon that keeps the Muon matrix update for hidden weight matrices while using a support-aware finite-cap linear minimization oracle for tied vocabulary tables and an RMS-normalized update for one-dimensional auxiliary parameters. AF-Muon therefore trains every parameter class with a single first-moment buffer and no second-moment state, saving around 20% optimizer-state memory relative to Hybrid Muon in our benchmark. Across nine tied-token settings—decoder-only language models from 124M to 1B parameters, a fully shared T5-style encoder–decoder, and ImageGPT-style image-token, protein, and sparse-MoE variants, spanning text, image, and protein-sequence data—AF-Muon improves mean validation loss and perplexity over both Hybrid Muon and a SCION-style Sign endpoint. Long-horizon runs and hyperparameter sensitivity studies confirm the gain is robust, and identical-momentum diagnostics attribute it to the finite cap, which preserves more within-row magnitude than Sign while bounding the coordinate concentration of row-RMS. These results identify tied vocabulary tables as a distinct optimizer geometry and yield a robust AdamW-free Muon variant across models, modalities, and architectures, with about 1% step-time overhead in matched training.

1 INTRODUCTION

Adam and AdamW remain standard for large-model training because their coordinate-wise first- and second-moment adaptation is robust across heterogeneous parameter types (Kingma & Ba, 2014; Loshchilov & Hutter, 2017). Muon replaces this geometry for selected two-dimensional hidden weights: it maps momentum through a short Newton–Schulz iteration approximating the polar factor, yielding a spectral-norm steepest-descent direction in the converged limit (Jordan et al., 2024). This matrix-aware update is effective at language-model scale, but the reference Muon recipe is still hybrid: embeddings, output heads, normalization parameters, and biases remain on an auxiliary AdamW branch (Liu et al., 2025). Thus Muon removes coordinate-wise second-moment adaptation from only part of the model, including neither the embedding nor output blocks that dominate token-model parameter count. This leaves a basic question: how should the remaining parameter roles be updated without AdamW while respecting their functional geometry?

Norm-based optimizer design emphasizes that tensor shape alone does not determine an appropriate update rule. Parameters with identical array shapes may implement different functions and therefore require different geometries (Bernstein & Newhouse, 2024; Large et al., 2024; Bernstein & Newhouse, 2025). Tied vocabulary tables provide a particularly sharp example. Weight tying uses one stored matrix $E \in \mathbb{R}^{V \times d}$ for both token lookup and output classification (Press & Wolf, 2017; Inan et al., 2016). The lookup path gives row-sparse gradients for observed input tokens, whereas the output-softmax path usually gives dense gradients over vocabulary rows (Press & Wolf, 2017; Lopardo et al., 2026). In fully shared encoder–decoder models, including recent models such as T5Gemma 2 (Zhang

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

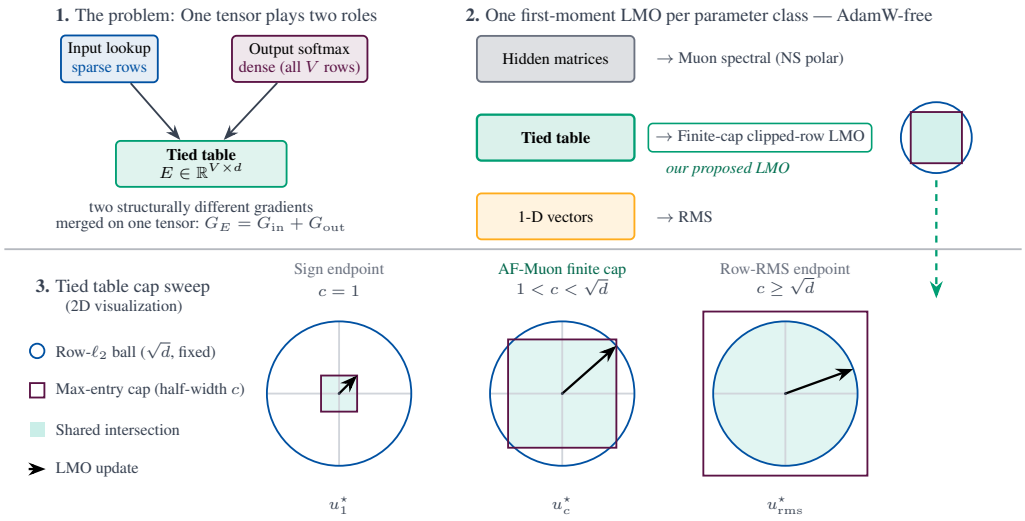


Figure 1: **AF-Muon tied-table geometry.** Top: a hard-tied vocabulary table serves both sparse input lookup and dense output-softmax roles, so their gradients merge on one tensor. AF-Muon assigns one first-moment LMO per parameter class: Muon for hidden matrices, a finite-cap clipped-row LMO for the tied table, and RMS normalization for 1-D vectors. Bottom: the tied-table feasible set is the intersection of a fixed row-ℓ_2 ball and a coordinate cap box. Varying c interpolates between the Sign endpoint ($c = 1$), the finite-cap AF-Muon update ($1 < c < \sqrt{d}$), and row-RMS ($c \geq \sqrt{d}$).

et al., 2025a), same table can receive separate encoder- and decoder-lookup contributions. Autograd merges these structurally different signals on one aliased tensor before the optimizer acts, so the tied table is neither an ordinary hidden linear map nor an isolated embedding or classifier.

Existing choices capture only parts of this structure. Hybrid Muon assigns the tied table to AdamW, retaining coordinate-wise second-moment state. SCION gives the closest role-aware LMO baseline: for untied one-hot inputs it recommends ColNorm–Spectral–Sign, where ColNorm corresponds to per-token row-RMS normalization in our $E \in \mathbb{R}^{V \times d}$ orientation; under input–output weight sharing it recommends Sign–Spectral–Sign, so the aliased table receives a Sign LMO (Pethick et al., 2025). We use this rule as our SCION-style endpoint baseline. Sign controls coordinate concentration but gives weak and dominant momentum coordinates equal magnitude; row-RMS preserves within-row magnitude information but can concentrate update energy in a few coordinates. Rescaling either endpoint changes only overall magnitude, leaving the geometric trade-off unresolved.

Figure 1 summarizes the tied-table problem, the AF-Muon parameter-role assignment, and the finite-cap geometry between the Sign endpoint and the uncapped row-RMS limit. We introduce *AF-Muon*, a role-complete AdamW-free extension of Muon for models with tied token tables. AF-Muon keeps Muon's spectral update for hidden matrices, uses RMS-normalized updates for one-dimensional auxiliary tensors, and applies a new support-aware finite-cap LMO to the merged momentum of the tied vocabulary table. The tied-table oracle intersects a row-wise RMS constraint with an entrywise cap, connecting Sign at the most restrictive cap to row-RMS when the cap is removed. Intermediate caps preserve more within-row magnitude information than Sign while limiting the coordinate concentration allowed by row-RMS. The cap c controls normalized direction geometry, while a separate scale s controls update magnitude. Thus every parameter class considered is trained with first-moment state only, with no parameter delegated to AdamW.

We evaluate AF-Muon under matched budgets across nine tied-token settings: decoder-only language models, a fully shared encoder–decoder model, image-token modeling, protein language modeling, and sparse MoE. AF-Muon achieves the lowest mean validation loss in every setting, outperforming both Hybrid Muon and the SCION-style Sign endpoint. Long-horizon runs, sensitivity studies, and matched-momentum diagnostics further support that the finite cap, rather than merely removing AdamW, drives the gain. Our contributions are:

- We identify and formulate optimization of a tied vocabulary table as an aliased multi-role problem: one stored tensor serves sparse lookup and dense classifier roles, so its merged gradient combines structurally different contributions.
- We derive an exact, support-aware finite-cap LMO that connects Sign and row-RMS while separating directional geometry, controlled by c, from update magnitude, controlled by s. Combined with Muon for hidden matrices and RMS-normalized updates for vectors, this yields an AdamW-free, second-moment-free optimizer that trains every parameter class with a single first-moment buffer, reducing optimizer-state memory relative to Hybrid Muon.
- We provide controlled evidence across nine tied-token settings, spanning 124M–1B-parameter decoder-only models plus encoder–decoder, image-token, protein-sequence, and sparse-MoE variants, trained on text, image, and protein data with three paired seeds, up to $\sim$10B tokens, and tied-geometry sensitivity studies.

2 METHOD

Muon (Jordan et al., 2024) approximately projects momentum updates for selected hidden matrices onto their polar factors, which, in the exact limit, corresponds to steepest descent over a spectral-norm ball. Common Hybrid-Muon implementations keep this matrix update for hidden weights but optimize embeddings, output heads, normalization parameters, and biases with an auxiliary AdamW branch (Loshchilov & Hutter, 2017). This works well in practice but mixes optimizer geometries and carries AdamW second-moment state for the auxiliary branch.

AF-Muon is a single modular optimizer that keeps Muon's hidden-matrix update while replacing the auxiliary AdamW branch with first-moment LMO updates matched to each parameter's role. In the tied-head RoPE (Rotary Position Embedding) architectures considered here, learned tensors are partitioned into hidden matrices, the shared tied token-embedding/LM-head table, and one-dimensional auxiliary tensors. Hidden matrices use the spectral LMO, vector parameters use an RMS LMO, and the tied table receives our new finite-cap LMO: since one stored table is aliased between token-lookup and output-classifier roles, we propose an intersection of a max-row-ℓ_2 ball and a max-entry ball for its merged update.

2.1 STEEPEST DESCENT UNDER A NORM

Let a parameter block $W \in \mathcal{W}$ carry a norm $\|\cdot\|$ with dual norm $\|Z\|_\star := \max_{\|U\|\le 1}\langle Z, U\rangle$, where $\langle A, B\rangle = \mathrm{tr}(A^\top B)$. For oracle input Z, a normalized steepest-descent direction, or *linear minimization oracle* (LMO), is

$$D_{\|\cdot\|}(Z) \in \arg\max_{\|U\|\le 1} \langle Z, U\rangle. \tag{1}$$

For $Z \neq 0$, $\langle Z, -\eta D_{\|\cdot\|}(Z)\rangle = -\eta\|Z\|_\star < 0$, so $-D_{\|\cdot\|}(Z)$ maximizes linearized decrease under $\|\cdot\|$. The tied table and the one-dimensional auxiliary tensors use the unnormalized momentum buffer $B_t = \mu B_{t-1} + G_t$ as their oracle input. Hidden matrices use Muon's Nesterov-style construction $M_t = \mu M_{t-1} + (1-\mu)G_t$, $Z_t = (1-\mu)G_t + \mu M_t$, with Z_t passed to the Newton–Schulz map. The identity therefore refers to the oracle input, not to each minibatch gradient G_t.

2.2 PARAMETER CLASSES

We partition stored parameters by functional role.

Hidden matrices. For hidden matrices $W \in \mathbb{R}^{m\times n}$, we use Muon's spectral LMO. If $Z_t = \Phi\Sigma\Psi^\top$, the exact spectral-norm LMO is the polar factor $\Phi\Psi^\top$ (Pethick et al. (2025)), approximated by K-step Newton–Schulz iteration, $\mathrm{NS}_K(Z_t)$. Following Muon, we apply the rectangular shape factor $\sqrt{\max(1, m/n)}$; this is the only branch to which we apply matrix **weight decay** in our experiments.

Vector parameters. For one-dimensional parameters (normalization gains, QK-norm gains, biases), the ideal RMS LMO is $D_{\mathrm{RMS}}(b) = b/\,\mathrm{RMS}(b)$ for $b \neq 0$ and 0 otherwise, where $\mathrm{RMS}(b) = \|b\|_2/\sqrt{d}$ for $b \in \mathbb{R}^d$. We implement the stabilized form $\widehat{D}_{\mathrm{RMS}}(b) = b/\max\{\mathrm{RMS}(b), \varepsilon\}$, which

equals D_{RMS} whenever $\mathrm{RMS}(b) \geq \varepsilon$. This removes AdamW from these parameters while preserving a native shape update.

Tied table (aliased). The tied token-embedding/LM-head is one stored tensor $E \in \mathbb{R}^{V\times d}$, where V is the vocabulary size and d is the hidden width. It used in two roles, $i \mapsto E_{i,:}$ and $h \mapsto Eh \in \mathbb{R}^V$; autograd accumulates a merged gradient $G^E = G^E_{\mathrm{in}} + G^E_{\mathrm{out}}$ on the shared tensor. Because the tied table receives a dense output-role gradient through the softmax, its LMO is applied to all rows. Unlike an untied head, the merged update should *not* be row-centered: shifting every row by $a^\top$ is a logit-preserving symmetry only for the output role, but under tying, it also shifts every token embedding, so the head-only quotient does not apply here (Appendix D) ; AF-Muon therefore applies its LMO to the merged, unprojected tied momentum.

More generally, if one stored tensor W is used in R roles through linear maps $A_r : \mathcal{W} \to \mathcal{X}_r$ with role norm N_r and radius ρ_r, pulling the product geometry back to W gives the *alias norm*

$$N_{\mathrm{alias}}(U) := \max_{1\leq r\leq R} \frac{N_r(A_r U)}{\rho_r}, \tag{2}$$

a norm whenever $\bigcap_r \ker A_r = \{0\}$ (Appendix C), whose unit ball is the intersection of the pulled-back role balls and whose LMO satisfies every role constraint at once. The tied table is the $R = 2$ case with $A_1 = A_2 = \mathrm{Id}$: a row-wise radius, reflecting lookup-row control, intersected with a Sign-compatible max-entry cap on the output side. This intersection geometry is a modeling choice, not a consequence of tying alone; we evaluate it against both endpoints.

For $U \in \mathbb{R}^{V\times d}$ with rows u_i, let $\|U\|_{\mathrm{row}} := \max_i \|u_i\|_2/\sqrt{d}$ and $\|U\|_{\max} := \max_{i,j} |U_{ij}|$. For cap $c \geq 1$, define the tied-table norm

$$\|U\|_{\mathrm{tie},c} := \max\left\{\|U\|_{\mathrm{row}},\ \|U\|_{\max}/c\right\}, \tag{3}$$

whose unit ball caps every row's ℓ_2 norm at $\sqrt{d}$ and every entry at c. We write $C_c = \{U : \|U\|_{\mathrm{tie},c} \leq 1\}$ for its unit ball. This feasible set is a Cartesian product across rows, so its LMO decomposes row-by-row. For a row momentum $b \in \mathbb{R}^d$ with support size $k(b)$, choose $\tau_c(b) \geq 0$ solving

$$\textstyle\sum_j \min\{c^2, \tau_c(b)^2 b_j^2\} = \min\{d, c^2 k(b)\}, \tag{4}$$

and define the *clipped-row LMO*

$$\mathcal{L}_c(b)_j = \mathrm{sign}(b_j)\min\{c, \tau_c(b)|b_j|\}. \tag{5}$$

Proposition 1 (Clipped-row LMO). *For $c \geq 1$, $\mathcal{L}_c(b) \in \arg\max\{\langle b, u\rangle : \|u\|_2 \leq \sqrt{d},\ \|u\|_\infty \leq c\}$, with $\|\mathcal{L}_c(b)\|_2^2 = \min\{d, c^2 k(b)\}$. The endpoints are $\mathcal{L}_1(b) = \mathrm{sign}(b)$ and, for $c \geq \sqrt{d}$, $\mathcal{L}_c(b) = b/\mathrm{RMS}(b)$ (row-RMS). We write $c = \infty$ as shorthand for omitting the coordinate cap.*

Applying $\mathcal{L}_c$ row-wise to the table momentum B gives a global LMO for $\|\cdot\|_{\mathrm{tie},c}$ (proof in Appendix E); under full-softmax training, momentum rows typically become dense enough that $c^2 k(b) \geq d$, in which case the selected direction has ℓ_2 norm $\sqrt{d}$. The cap c thus controls the normalized *shape* of the update within each row, interpolating between Sign and row-RMS, while a separate scalar s (equivalently, the scaled feasible set $s\,C_c$) controls per-row *magnitude* without changing that shape.

The finite cap trades symmetry for concentration control: $\mathcal{L}_c$ is equivariant under row and signed-feature permutations , but, unlike row-RMS, not under general right rotations $R \in O(d)$, unavoidably so, since any right-orthogonally equivariant, fixed-energy map that is positively aligned with its input must be exactly row-RMS, with scale invariance following automatically (Appendix G) Concretely, for dense rows ($c^2 k(b) \geq d$) the peak-to-average ratio $\mathrm{PAPR}(u) := \|u\|_\infty^2/(\|u\|_2^2/d)$ obeys $\mathrm{PAPR}(\mathcal{L}_c(b)) \leq c^2$, and optimal linear objective is nondecreasing in c from Sign to row-RMS (Appendix F): finite-cap family traces an alignment–concentration frontier between two endpoints.

As with the other non-hidden-matrix blocks, the tied table accumulates $B_t^E = \mu B_{t-1}^E + G_t^E$ from the merged gradient G_t^E. For the tied table, $O_{t,i} = \mathcal{L}_c(\mathrm{row}_i(B_t^E))$ and $E_t = E_{t-1} - \eta_{\mathrm{tie},t} O_t$. We implement the scalar tied multiplier as a width-scaled learning rate: with shared schedule multiplier a_t, $\eta_{\mathrm{M},t} = \eta_{\mathrm{M}}^{\max} a_t$, and

$$\eta_{\mathrm{tie},t} = \eta_{\mathrm{M},t}\,(\rho_{\mathrm{out}}/\rho_{\mathrm{hid}})\,s/d. \tag{6}$$

In our experiments $\eta_{\mathrm{M}}^{\max} = 0.02$ and $\rho_{\mathrm{out}}/\rho_{\mathrm{hid}} = 3000/50 = 60$, giving $\eta_{\mathrm{tie},t} = 1.2\,a_t s/d$; our main setting is $c = 3$, $s = 0.5$, which for all evaluated widths ($d > 9$) lies strictly between Sign and row-RMS.

When does sharing require a new geometry? Not always. If one dense matrix is reused at L depths under the *same* spectral constraint, the intersection of identical balls collapses back to that ball, so the right update is the ordinary spectral LMO $D_{\|\cdot\|_{\mathrm{op}}}$ applied once to the accumulated gradient, $D_{\|\cdot\|_{\mathrm{op}}}(\sum_\ell G_\ell)$. The tied vocabulary table differs because its lookup and classifier aliases impose genuinely heterogeneous row-wise and entrywise constraints whose intersection does not collapse: in a four-depth shared-MLP diagnostic, combining per-depth Muon directions after matching spectral radius retains only 74.1% (up-projection) and 83.9% (down-projection) of the linear objective (Appendix K.9).

2.3 Full optimizer

Algorithm 1 combines these branches into a single AdamW-free optimizer, using the oracle inputs of §2.1: the Nesterov construction for hidden matrices, and the unnormalized buffer $B_t = \mu B_{t-1} + G_t$ for the tied table and auxiliary tensors. Decoupled weight decay applies only to hidden matrices; the tied table and auxiliary tensors receive none. The row oracle $\mathcal{L}_c$ uses closed-form fast paths at $c = 1$ and $c \geq \sqrt{d}$; for $1 < c < \sqrt{d}$ we solve equation 4 by per-row bisection in FP32, in row-preserving chunks so that chunking never changes the computed update. The algorithm is stated for tied-head RoPE decoders, our headline setting; untied heads and learned positional tables are not part of the tied-table claim, and in GPT-style diagnostics with learned positional embeddings we keep WPE on the fixed baseline Sign rule rather than claim a new positional-table optimizer.

Algorithm 1: AF-Muon with tied-table clipped LMO

Require: LR schedule a_t, peak Muon LR $\eta_{\mathrm{M}}^{\max}$, peak vector LR $\eta_{\mathrm{vec}}^{\max}$, momentum μ, matrix weight decay λ_{M}, tied update scale s, tied cap c, output radius ρ_{out}, hidden radius ρ_{hid}, Newton–Schulz steps K, constant ε

Let $\mathcal{P}$ be the set of unique stored parameter tensors, deduplicated by object identity
Initialize hidden-matrix momentum buffers $M_0^W \leftarrow 0$
Initialize auxiliary momentum buffers $B_0^W \leftarrow 0$
for step $t = 1, \ldots, T$ **do**
  $\eta_{\mathrm{M},t} \leftarrow \eta_{\mathrm{M}}^{\max} a_t$; $\eta_{\mathrm{vec},t} \leftarrow \eta_{\mathrm{vec}}^{\max} a_t$
  Compute gradients $G_t^W \leftarrow \nabla_W \mathcal{L}_t(\theta_{t-1})$
  for each $W \in \mathcal{P}$ **do**
    if G_t^W is `None` **then**
      continue
    end if
    if W is a hidden matrix in $\mathbb{R}^{m\times n}$ **then**
      $M_t^W \leftarrow \mu M_{t-1}^W + (1-\mu) G_t^W$
      $Z_t^W \leftarrow (1-\mu) G_t^W + \mu M_t^W$
      $O_t^W \leftarrow \mathrm{NS}_K(Z_t^W) \cdot \sqrt{\max(1, m/n)}$
      $W_t \leftarrow (1 - \eta_{\mathrm{M},t}\lambda_{\mathrm{M}}) W_{t-1} - \eta_{\mathrm{M},t} O_t^W$
    else if $W \in \mathbb{R}^{V\times d}$ is the tied token embedding/ LM-head table **then**
      G_t^W *is the shared tensor gradient, containing both input-lookup and output-softmax contributions*
      $B_t^W \leftarrow \mu B_{t-1}^W + G_t^W$
      $\eta_{\mathrm{tie},t} \leftarrow \eta_{\mathrm{M},t} (\rho_{\mathrm{out}}/\rho_{\mathrm{hid}})\, s/d$
      for each row $i = 1, \ldots, V$ **do**
        $b_i \leftarrow \mathrm{row}_i(B_t^W)$
        $u_i \leftarrow \mathcal{L}_c(b_i)$
      end for
      $O_t^W \leftarrow U$, where $\mathrm{row}_i(U) = u_i$
      $W_t \leftarrow W_{t-1} - \eta_{\mathrm{tie},t} O_t^W$
    else if W is a one-dimensional auxiliary tensor **then**
      $B_t^W \leftarrow \mu B_{t-1}^W + G_t^W$
      $O_t^W \leftarrow B_t^W / \max\{\mathrm{RMS}(B_t^W), \varepsilon\}$
      $W_t \leftarrow W_{t-1} - \eta_{\mathrm{vec},t} O_t^W$
    else
      error: every parameter must be assigned an explicit role
    end if
  end for
end for
return θ_T

AF-Muon differs from Hybrid Muon in two ways: no parameter class is delegated to AdamW or carries second-moment state, and the tied table is not treated as a generic auxiliary tensor but receives the clipped intersection LMO above, motivated by its dual input-lookup/output-softmax role. While we establish exact oracle properties and conditional convergence bounds for idealized variants of AF-Muon in Appendix H, we emphasize that the practical implementation relies on momentum and approximate oracle maps (including finite-step Newton–Schulz and numerical bisection). Consequently, we make no claim of unconditional convergence for this stochastic setting.

Relationship to prior norm-based optimizers. AF-Muon remains within the norm-based/LMO optimizer family (Large et al., 2024; Jordan et al., 2024; Pethick et al., 2025; Riabinin et al., 2025). We claim no novelty for Muon's spectral update or for the existence of Sign or row-RMS LMOs. Our contribution is the tied-table geometry: when input and output embeddings are tied, the stored tensor is aliased across two roles, and we propose the clipped row-normalizer family equation 5, contains the standard endpoints as special cases but exposes finite interior updates, evaluated in Section 3.

Table 1: Main optimizer comparison across tied-token models and domains. Results are mean ± SD over three paired seeds. All rows compare matched tied-table optimizer arms under the same model, data, token budget, precision, batching, and matrix-update protocol; detailed settings and per-seed results are reported in Appendix J. NanoGPT, SmolLM2, Qwen2.5, Llama, T5, MoE, and protein rows report validation perplexity; ImageGPT reports validation bits/dim.

Model / setting	Optimizer	Train loss ↓	Val. loss ↓	Val. PPL / bpd ↓
NanoGPT	Hybrid Muon	3.5150 ± 0.0028	3.5177 ± 0.0032	33.71 ± 0.11
	SCION-style Sign	3.5125 ± 0.0107	3.5188 ± 0.0124	33.75 ± 0.42
	AF-Muon	**3.4898 ± 0.0011**	**3.4919 ± 0.0019**	**32.85 ± 0.06**
SmolLM2-135M	Hybrid Muon	3.5088 ± 0.0028	3.5065 ± 0.0021	33.33 ± 0.07
	SCION-style Sign	3.4727 ± 0.0069	3.4706 ± 0.0040	32.16 ± 0.13
	AF-Muon	**3.4453 ± 0.0006**	**3.4446 ± 0.0036**	**31.33 ± 0.11**
Qwen2.5-0.5B	Hybrid Muon	3.6269 ± 0.0074	3.6393 ± 0.0039	38.06 ± 0.15
	SCION-style Sign	3.5984 ± 0.0046	3.6109 ± 0.0031	37.00 ± 0.11
	AF-Muon	**3.5713 ± 0.0052**	**3.5831 ± 0.0026**	**35.99 ± 0.09**
Llama-3.2-1B	Hybrid Muon	3.4932 ± 0.0027	3.5388 ± 0.0035	34.42 ± 0.12
	SCION-style Sign	3.4771 ± 0.0033	3.5225 ± 0.0075	33.87 ± 0.25
	AF-Muon	**3.4594 ± 0.0027**	**3.5056 ± 0.0048**	**33.30 ± 0.16**
Small T5-style	Hybrid Muon	5.3124 ± 0.0050	5.3374 ± 0.0102	207.98 ± 2.12
	SCION-style Sign	5.1477 ± 0.0080	5.1763 ± 0.0150	177.03 ± 2.66
	AF-Muon	**5.1476 ± 0.0139**	**5.1708 ± 0.0034**	**176.06 ± 0.61**
T5Gemma2-style	Hybrid Muon	3.6409 ± 0.0044	3.6548 ± 0.0041	38.66 ± 0.16
	SCION-style Sign	3.5698 ± 0.0011	3.5859 ± 0.0017	36.09 ± 0.06
	AF-Muon	**3.5573 ± 0.0010**	**3.5735 ± 0.0010**	**35.64 ± 0.04**
NanoGPT-MoE	Hybrid Muon	3.4873 ± 0.0031	3.4902 ± 0.0019	32.79 ± 0.06
	SCION-style Sign	3.5106 ± 0.0033	3.5129 ± 0.0025	33.55 ± 0.08
	AF-Muon	**3.4662 ± 0.0011**	**3.4677 ± 0.0007**	**32.06 ± 0.02**
ImageGPT-style	Hybrid Muon	2.3236 ± 0.0067	2.3346 ± 0.0049	1.1227 ± 0.0024
	SCION-style Sign	2.3568 ± 0.0049	2.3712 ± 0.0055	1.1403 ± 0.0027
	AF-Muon	**2.3208 ± 0.0101**	**2.3243 ± 0.0055**	**1.1177 ± 0.0026**
Protein LM	AdamW	7.951 ± 0.023	7.925 ± 0.091	2772.6 ± 258.2
	Hybrid Muon	7.478 ± 0.013	6.874 ± 0.008	967.0 ± 7.9
	SCION-style Sign	7.464 ± 0.015	6.816 ± 0.012	912.7 ± 11.3
	AF-Muon	**7.426 ± 0.019**	**6.760 ± 0.019**	**862.3 ± 16.5**

3 Experiments

3.1 Experimental setup

We evaluate AF-Muon on nine tied-token settings: four decoder-only language models on FineWeb, two shared-vocabulary encoder–decoder configurations, sparse NanoGPT-MoE, ImageGPT-style ImageNet-32 token modeling, and protein language modeling on UniRef50. Unless otherwise stated, each optimizer comparison uses three paired seeds with the same model, tokenizer or token cache, data order, token budget, batching, precision, evaluation shard, and matrix-update protocol. All Muon-family runs use Muon's spectral update for hidden matrices with matrix weight decay 0.1, FP32 trainable parameters and optimizer state, BF16 autocast, and gradient clipping at norm 1.0. Hybrid Muon keeps embeddings, tied tables, normalization parameters, and biases on AdamW, with auxiliary weight decay 0.01 where used. SCION-style Sign replaces the tied table by the Sign endpoint $(c, s) = (1, 1)$. AF-Muon uses the tied-table clipped LMO from Section 2.2 with $(c, s) = (3, 0.5)$, peak tied-table learning rate $\eta_{\text{tie}}^{\max} = 1.2s/d$, and scheduled rate $\eta_{\text{tie},t} = a_t \eta_{\text{tie}}^{\max}$. For the seven two-role tied-table settings, the output-to-hidden radius ratio is $\rho_{\text{out}}/\rho_{\text{hid}} = 3000/50 = 60$ (Pethick et al., 2025), matching the SCION-style output-layer scaling used for Sign relative to Spectral updates. The T5Gemma2-style experiment uses the main $\rho_{\text{out}}/\rho_{\text{hid}} = 60$ setting but for the small fully shared T5-style table, which receives encoder-input, decoder-input, and output-classifier gradients, we additionally sweep this ratio and use $\rho_{\text{out}}/\rho_{\text{hid}} = 1$; details are reported in Appendix J.5.

3.2 Main decoder-only language modeling results

Table 1 compares Hybrid Muon with the best current AF-Muon configuration on each model. Across all settings, AF-Muon achieves the best mean validation loss and PPL/BPD. Paired three-seed tests against the strongest non-AF baseline are reported with the per-seed results in Appendix J.

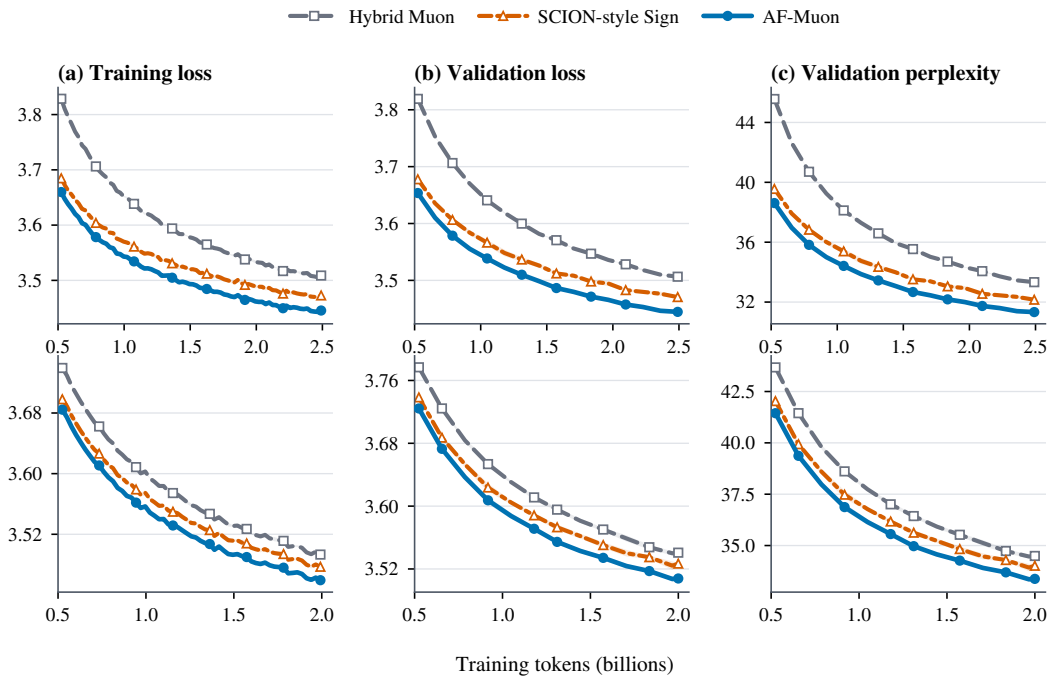


Figure 2: Decoder-only tied-vocabulary training dynamics on FineWeb for SmolLM2-135M (top) and Llama-3.2-1B (bottom). Each row reports metrics under the matched protocol in Section 3.1. Curves show the mean over three paired seeds. Llama model long horizon curves reported in Appendix J.4.1

NanoGPT. NanoGPT provides the cleanest controlled decoder-only test of the aliased sparse–dense tied table. AF-Muon obtains the best validation loss and perplexity across all three paired seeds, showing that replacing the auxiliary AdamW treatment of the shared table with the finite-cap LMO improves the matched Muon matrix-update protocol. long horizon curves and details are provided in Appendix J.1 Ablations over gradient source, batch size, auxiliary learning rate, vocabulary size, model width, and cap/scale choices are reported in Appendix K.

SmolLM2-135M. SmolLM2-135M tests the same tied-table update in a pretrained-family decoder-only architecture at larger width and depth. At 2.49B training tokens, AF-Muon achieves the best validation loss and perplexity across all paired seeds, showing that the finite-cap tied-table update remains useful beyond the controlled NanoGPT architecture. Details are reported in Appendix J.2.

Qwen2.5-0.5B. Qwen2.5-0.5B tests the tied-table update in a larger decoder-only architecture with a substantially larger shared vocabulary table. At the near-1B evaluation point, AF-Muon achieves the best validation loss and perplexity across all paired seeds. A seed-43 long-horizon continuation shows the same stable ordering later in training, with details reported in Appendix J.3, and J.3.1.

Llama-3.2-1B. Llama-3.2-1B is the largest decoder-only tied-vocabulary setting in our main comparison. Near the 2B-token budget, AF-Muon again achieves the best validation loss and perplexity across all paired seeds. A seed-43 continuation to 9.98B tokens provides a longer-horizon stability check and preserves the same ordering; full results are reported in Appendix J.4, and J.4.1.

3.3 Generalization beyond dense decoder-only language modeling

Encoder–decoder shared vocabulary. Motivated by T5Gemma 2 (Zhang et al., 2025a), we use two experiments to ask whether AF-Muon transfers to three-way shared vocabularies and whether it remains effective in a larger configuration. In the small T5-style control (275M tokens, 49k shared vocabulary, AF-Muon and SCION-style Sign both improve over tied Muon. In T5Gemma2-style (786M parameters, 262k shared vocabulary, 750M target tokens), AF-Muon gives the best validation loss in the evaluated seeds. Together, these results show that AdamW-free shared-table optimization transfers to three-way tying. Full results are in Appendix J.5.

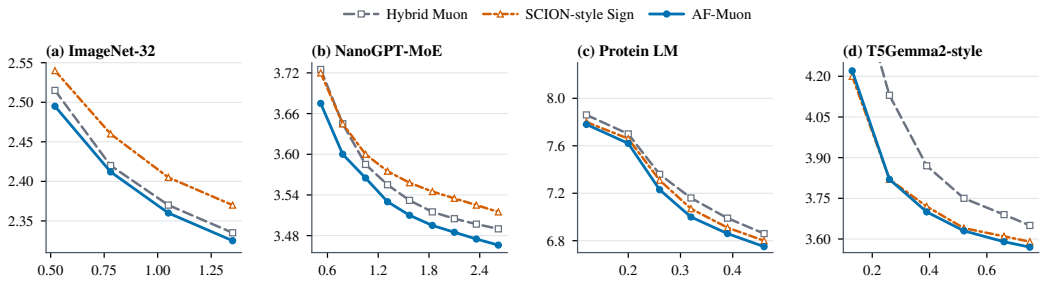


Figure 3: Validation-loss curves beyond decoder-only language modeling. Each panel shows one tied-token setting with its own training-token scale. AF-Muon reaches the lowest final validation loss in all four settings; full training-loss and perplexity curves are in the related appendix.

Sparse NanoGPT-MoE. following the MoE scaling direction of recent models such as DeepSeek-V3 and OLMoE (Liu et al., 2024; Muennighoff et al., 2025), and Muon-family optimizers have been used at frontier MoE scale in Kimi K2 (Team et al., 2025). We replace NanoGPT MLPs with four top-1 routed experts while keeping the vocabulary table tied. At 2.62B tokens, AF-Muon achieves the best validation loss and perplexity; Sign underperforms Muon, indicating that the finite cap, not merely removing AdamW, accounts for the improvement. Full details are in (Appendix J.6).

ImageGPT-style ImageNet-32. Autoregressive discrete-token modeling underlies modern image and image–text generators, including LlamaGen, VAR, Chameleon, and Emu3 (Sun et al., 2024; Tian et al., 2024; Team, 2024; Wang et al., 2024). Motivated by this discrete-token modeling paradigm, we test an ImageGPT-style causal transformer on ImageNet-32 with an intentionally tied input/output color-token table. Images are represented as 1024 RGB554 tokens from a 16k vocabulary, so the shared table again receives sparse lookup and dense classifier gradients. At 1.31B image tokens, AF-Muon achieves the best validation loss and bits/dim, and a fixed top-half completion evaluation gives the same ordering. Full settings, and results are reported in Appendix J.7.

Protein language modeling. Autoregressive protein language models such as ProtGPT2 and ProGen2 model protein sequences with GPT-style next-token objectives (Ferruz et al., 2022; Nijkamp et al., 2023), letting us test the aliased sparse-lookup/dense-classifier tied-table structure on a non-linguistic token distribution. Following ProtGPT2, we use its BPE tokenizer and train a randomly initialized 363M-parameter RoPE decoder with tied token embedding / LM head on the UniRef50 2021_04 release (Suzek et al., 2015). At 458.8M training tokens, AF-Muon achieves the best metrics, and all Muon-family arms substantially outperform tuned AdamW. Full details are in Appendix J.8.

3.4 Ablations and Robustness

Causal attribution and scope. A 2×2 factorial ablation independently replaces the tied-table AdamW update with the finite-cap LMO and the one-dimensional AdamW update with the RMS-LMO (Figure 4;Appendix K.1). Both substitutions contribute, and the joint improvement is larger than the sum of the isolated effects, indicating an interaction between the two role-specific updates. A complementary gradient-source intervention shows why the tied table requires mixed-role treatment: the dense output component dominates, but adding the sparse input component still improves validation, and the two components are nearly orthogonal (cosine 0.003) (Appendix K.8). Conversely, a negative boundary control shows that sharing alone is insufficient: when one dense matrix is reused at multiple depths, every use induces the same spectral geometry, so their intersection collapses to the original spectral ball and Muon should be applied once to the accumulated gradient (Appendix K.9).

Geometry versus update scale. At fixed $c = 3$, every tested scale $s \in [0.125, 0.5]$ achieves lower validation loss than the standard SCION-style Sign arm in the single-seed sensitivity sweep, with a broad favorable region around intermediate scales (Appendix K.3). A joint sweep over $c \in \{1, 2, 3, 4, 6, 10, \infty\}$ and $s \in \{0.5, 1\}$ further shows that performance is more sensitive to s than to the precise cap: at $s = 0.5$, validation losses span only 0.0069 across the complete cap range (Appendix K.7). We therefore do not present $c = 3$ as a uniquely optimal post-hoc choice. Rather, it realizes a nondegenerate interior geometry: in this audit it caps only 0.29% of coordinates while affecting 90.3% of rows and attaining $1.253\times$ the same-state Sign linear objective. The sweep also recovers the expected Sign and row-RMS endpoint behavior at $c = 1$ and $c = \infty$, respectively.

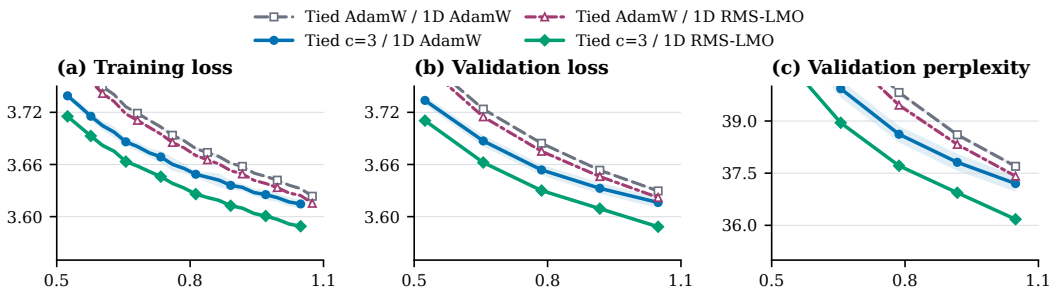


Figure 4: NanoGPT 2×2 factorial ablation over training tokens (billions) for three paired seeds.

Robustness across scales and protocols. The AF-Muon advantage over Sign persists across genuinely re-tokenized 50k, 100k, and 150k vocabularies, with within-tokenizer mean validation-loss improvements of 0.036, 0.045, and 0.052 (Appendix K.2). It also persists at real hidden widths $d \in \{768, 1024, 1536\}$, where the three-seed paired improvements are 0.036, 0.040, and 0.032 (Appendix K.4). In single-seed protocol sweeps, AF-Muon obtains the lowest observed validation loss at every tested batch size from 32k to 524k tokens per update (Figure 5;Appendix K.5) and at every point in a $16\times$ auxiliary-branch learning-rate grid (Appendix K.6). We treat these latter sweeps as sensitivity evidence rather than multi-seed uncertainty estimates.

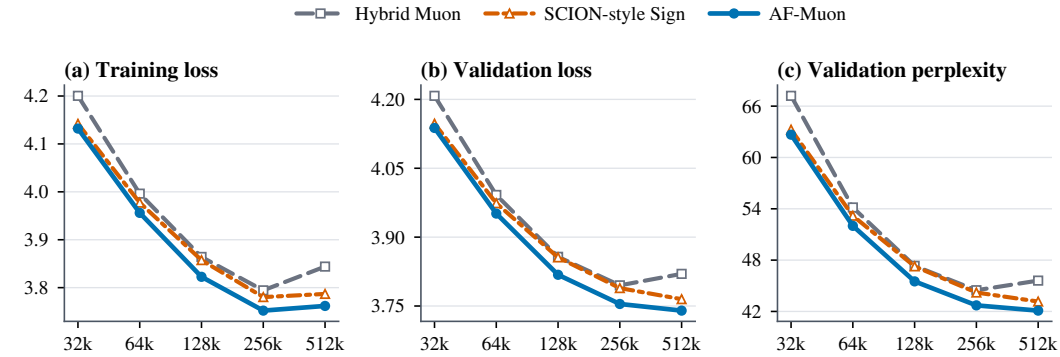


Figure 5: NanoGPT batch-size sensitivity at a matched 458.8M-token comparison. The x-axis varies the number of tokens per optimizer update. AF-Muon leads across all tested batch sizes.

4 Limitations

Weight tying does not uniquely imply the intersection of max-row-ℓ_2 and max-entry constraints: we prove that our update is the exact support-aware LMO for this proposed feasible set, not that the set is optimal among all possible role-aware geometries. The default $c = 3$, $s = 0.5$ is a robust configuration rather than a universally optimal choice. Our results establish robustness under matched fixed-token protocols; they do not tune every optimizer family or test frontier-scale pretraining.

5 Conclusion

We introduced AF-Muon, an AdamW-free extension of Muon that treats the tied vocabulary table as an *aliased* parameter block—one stored matrix receiving sparse lookup and dense classifier gradients—rather than a generic auxiliary tensor. It keeps Muon's spectral update for hidden matrices, RMS updates for one-dimensional parameters, and assigns the tied table the exact support-aware LMO of an intersection of max-row-ℓ_2 and max-entry constraints, removing all second-moment state and cutting optimizer memory by about 20% at roughly 1% training-time overhead. Across nine tied-token settings spanning 124M–1B-parameter decoder-only, fully shared encoder–decoder, image-token, protein, and sparse-MoE, trained on text, image, and protein data up to $\sim$10B tokens, AF-Muon improves fixed-budget validation over Muon and the SCION-style Sign endpoint under matched three-seed protocols, with ablations and sensitivity studies. These results support treating functional aliasing, not stored tensor shape alone, as a first-class consideration in optimizer design.

AI USE STATEMENT

We used generative AI tools for manuscript polishing, LaTeX debugging, code debugging, and internal consistency checks. All experiments, results, claims, and final artifacts were reviewed and verified by the authors, who take full responsibility for the content of the paper.

ETHICS STATEMENT

This work studies optimizer design for tied-token models and does not involve human subjects, private user data, or the release of a new dataset. Experiments use public or standard research datasets and model families for language, image-token, and protein-sequence modeling. The main ethical considerations are computational cost, reproducibility, and responsible reporting of empirical claims. We therefore report matched-seed comparisons, ablations, implementation details, and limitations, and we avoid claims about downstream deployment or societal impact beyond the optimizer behavior studied here.

REPRODUCIBILITY STATEMENT

We support reproducibility by reporting the full optimizer specification in Algorithm 1, the tied-table geometry and derivations in the appendix, and matched experimental settings in Section 3.1. The appendix provides per-seed results, ablations, diagnostics, overhead measurements, and additional training curves for the model families and domains studied in the main paper. We provide anonymized source code, configuration files, plotting scripts, and instructions for reproducing the reported runs in the supplementary material.

REFERENCES

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology, 2024. URL https://arxiv.org/abs/2409.20325.

Jeremy Bernstein and Laker Newhouse. Modular duality in deep learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3920–3930. PMLR, 2025. URL https://proceedings.mlr.press/v267/bernstein25a.html.

Francesco Bertolotti and Walter Cazzola. By tying embeddings you are assuming the distributional hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 3584–3610. PMLR, 2024. URL https://proceedings.mlr.press/v235/bertolotti24a.html.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. ISBN 978-0-521-83378-3.

Da Chang, Yongxiang Liu, and Ganzhao Yuan. On the convergence of muon and beyond. *CoRR*, abs/2509.15816, 2025. doi: 10.48550/ARXIV.2509.15816. URL https://doi.org/10.48550/arXiv.2509.15816.

Da Chang, Qiankun Shi, Lvgang Zhang, Di He, Yaoshuai Ma, Ganzhao Yuan, and Yongxiang Liu. Muse: Representation geometry of Muon beyond normalized momentum, 2026. URL https://arxiv.org/abs/2607.14536.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms. In *Advances in Neural Information Processing Systems*, volume 36, pp. 49205–49233, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/9a39b4925e35cf447ccba8757137d84f-Abstract.html.

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13(4348), July 2022. doi: 10.1038/s41467-022-32007-7. URL https://doi.org/10.1038/s41467-022-32007-7.

Oleg Filatov, Jiangtao Wang, Jan Ebert, and Stefan Kesselheim. Optimal scaling needs optimal norm, 2025. URL https://arxiv.org/abs/2510.03871.

Athanasios Glentis, Jiaxiang Li, Andi Han, and Mingyi Hong. Memory-efficient llm pretraining via minimalist optimizer design. *arXiv preprint arXiv:2506.16659*, 2025.

Kristina Gulordava, Laura Aina, and Gemma Boleda. How to represent a word and predict it, too: Improving tied architectures for language modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2936–2941. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1323. URL https://aclanthology.org/D18-1323/.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1842–1850. PMLR, 2018. URL https://proceedings.mlr.press/v80/gupta18a.html.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

Hakan Inan, Khashayar Khosravi, and Richard Socher. Tying word vectors and word classifiers: A loss framework for language modeling. *arXiv preprint arXiv:1611.01462*, 2016.

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 427–435, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/jaggi13.html.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. https://kellerjordan.github.io/posts/muon/, 2024. Accessed: 2026-07-05.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Tim Large, Yang Liu, Minyoung Huh, Hyojin Bahng, Phillip Isola, and Jeremy Bernstein. Scalable optimization in the modular norm. *Advances in Neural Information Processing Systems*, 37: 73501–73548, 2024.

Tim Tsz-Kit Lau and Weijie Su. Symmetry-compatible principle for optimizer design: Embeddings, lm heads, swiglu mlps, and moe routers. *arXiv preprint arXiv:2605.18106*, 2026.

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

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

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

Antonio Lopardo, Avyukth Harish, Catherine Arnett, and Akshat Gupta. Weight tying biases token embeddings towards the output space. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40795–40810. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.2027. URL https://aclanthology.org/2026.findings-acl.2027/.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. OLMoE: Open mixture-of-experts language models. In *International Conference on Learning Representations*, volume 2025, pp. 62061–62121, 2025.

Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: Exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

Saurabh Page, Advait Joshi, and SS Sonawane. Muonall: Muon variant for efficient finetuning of large language models. *arXiv preprint arXiv:2511.06086*, 2025.

Nikolaos Pappas, Lesly Miculicich, and James Henderson. Beyond weight tying: Learning joint input–output embeddings for neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 73–83. Association for Computational Linguistics, 2018. doi: 10.18653/v1/W18-6308. URL https://aclanthology.org/W18-6308/.

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained LMOs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 49069–49104. PMLR, 2025. URL https://proceedings.mlr.press/v267/pethick25a.html.

Thomas Pethick, Wanyun Xie, Roman Machacek, and Volkan Cevher. Optimistic dual averaging unifies modern optimizers. *arXiv preprint arXiv:2605.11172*, 2026.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 157–163. Association for Computational Linguistics, 2017. doi: 10.18653/v1/E17-2025. URL https://aclanthology.org/E17-2025/.

Xun Qian and Peter Richtárik. Convergence analysis of muon-type methods with inexact lmo in the degenerate case. *arXiv preprint arXiv:2606.21581*, 2026.

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.

R. Tyrrell Rockafellar. *Convex Analysis*, volume 28 of *Princeton Mathematical Series*. Princeton University Press, Princeton, NJ, 1997. ISBN 9780691015866.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4596–4604. PMLR, 2018. URL https://proceedings.mlr.press/v80/shazeer18a.html.

Chongjie Si, Debing Zhang, and Wei Shen. AdaMuon: Adaptive Muon optimizer. *arXiv preprint arXiv:2507.11005*, 2025.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Baris E. Suzek, Yuqi Wang, Hongzhan Huang, Peter B. McGarvey, and Cathy H. Wu. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, March 2015. doi: 10.1093/bioinformatics/btu739. URL https://doi.org/10.1093/bioinformatics/btu739.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi K2: Open agentic intelligence, 2025.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Biao Zhang, Paul Suganthan, Gaël Liu, Ilya Philippov, Sahil Dua, Ben Hora, Kat Black, Gus Martins, Omar Sanseviero, Shreya Pathak, et al. T5Gemma 2: Seeing, reading, and understanding longer. *arXiv preprint arXiv:2512.14856*, 2025a.

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik Durk Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more. In *International Conference on Learning Representations*, volume 2025, pp. 28033–28063, 2025b.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

CONTENTS

A Related Work **15**

B Setting, notation, and standard LMO geometry **17**

B.1 Notations 18

B.2 Dual norms and linear minimization oracles 18

C The alias-pullback framework **19**

C.1 Where does the terminology come from? 19

C.2 Pullback norm and intersection ball 19

C.3 Dual of the alias geometry 20

C.4 Redundancy under aliasing 21

D Hard-tied token tables **21**

D.1 Gradient structure of the tied table 21

D.2 Why the untied-head quotient disappears 22

D.3 The tied intersection norm 23

E Exact support-aware clipped-row oracle **23**

F Dual geometry, objective retention, and concentration control **25**

F.1 The tied dual norm as an infimal convolution 25

F.2 Explicit row support function and objective retention 26

F.3 Support-aware concentration (PAPR) bounds 27

G Symmetry and its limitation **28**

H Exact oracle properties and conditional convergence bounds **29**

H.1 Product geometry 29

H.2 Exact deterministic descent 30

H.3 Master inexact-oracle theorem 30

H.4 Decoupled decay as conditional gradient 31

H.5 Hybrid stationarity 32

I Extension to Encoder-Decoder Architectures (T5) **33**

I.1 Gradient Structure and Convergence 33

J Details of Experiments **34**

J.1 NanoGPT 34

J.2 SmolLM2-135M 35

J.3 Qwen2.5-0.5B 36

J.4 Llama-3.2-1B Decoder-only Results 38
J.5 Shared-vocabulary encoder–decoder experiments 42
J.6 NanoGPT-MoE 44
J.7 Tied ImageGPT-style ImageNet-32 45
J.8 Protein Language Modeling 50
J.9 Training Overhead 51

K Ablations **52**
K.1 NanoGPT Factorial Ablation: Tied Table Versus 1D Auxiliary Parameters 52
K.2 Real Vocabulary-Size Re-tokenization Ablation 52
K.3 Tied-Step Magnitude Ablation 54
K.4 NanoGPT Real-Width Ablation 55
K.5 Batch-size Sensitivity 57
K.6 Auxiliary Learning-rate Sensitivity 57
K.7 AF-Muon Cap and Scale Sensitivity 58
K.8 Sparse-input and Dense-output Gradient Sources 59
K.9 Boundary case: homogeneous dense-parameter reuse 60

A Related Work

Hybrid Muon and auxiliary updates. Adam and AdamW use coordinate-wise exponential moving averages of first and second moments and remain strong default optimizers for language-model pretraining (Kingma & Ba, 2014; Loshchilov & Hutter, 2017). Related methods modify this trade-off in different ways: Adafactor factorizes matrix second moments to reduce persistent state (Shazeer & Stern, 2018), Shampoo uses tensor-mode gradient statistics to construct structured preconditioners (Gupta et al., 2018), and Lion removes the second moment but applies a coordinate-wise Sign update to an interpolation of current and past gradients (Chen et al., 2023). Muon instead maps momentum for selected two-dimensional hidden weights through a short Newton–Schulz iteration that approximates the polar factor, i.e., the exact spectral-norm steepest-descent direction in the converged limit (Jordan et al., 2024). The reference Muon recipe explicitly leaves input embeddings, output heads, gains, and biases to AdamW, and subsequent scaling work emphasizes that weight decay and parameter-class update scales are important at language-model scale (Liu et al., 2025). At frontier scale, Muon underlies trillion-parameter training: Kimi K2 pretrains a one-trillion-parameter mixture-of-experts model with a Muon-based optimizer and a QK-clipping mechanism that suppresses attention-logit explosion (Team et al., 2025), underscoring that the output-side and stability behavior of Muon-style updates remains a live concern at scale. AF-Muon preserves Muon's hidden-matrix branch but replaces the entire auxiliary AdamW branch with one-buffer, role-specific updates.

Norm- and LMO-based design. The norm-based view treats an update rule as a choice of geometry rather than as a shape-agnostic transformation of coordinates. In particular, Bernstein & Newhouse (2024) argue that tensors with the same array shape can require different norms when they implement different functions, such as linear maps and embedding lookups. The modular norm recursively constructs an architecture-level geometry and studies hyperparameter transfer across model scale (Large et al., 2024), while modular duality assigns operator norms from layer semantics and derives efficient duality maps for Embed, Linear, and Conv2D modules (Bernstein & Newhouse, 2025). SCION develops stochastic optimization with norm-ball linear minimization oracles (LMOs), applies them to unconstrained neural-network training, and studies transferable layer radii (Pethick et al., 2025). Gluon develops a layerwise LMO framework and generalized-smoothness model that more closely reflects practical Muon and SCION implementations (Riabinin et al., 2025). A unifying view casts Muon, Lion, and related rules as instances of optimistic dual averaging (Pethick

et al., 2026), and a growing body of analysis establishes convergence for Muon-type methods under non-Euclidean and inexact-oracle assumptions (Chang et al., 2025; Qian & Richtárik, 2026). Recent scaling evidence also finds the output layer to be especially sensitive to its group learning rate and identifies its operator norm as a transferable quantity (Filatov et al., 2025). AF-Muon follows this lineage but separates two choices for the tied table: the cap c selects its normalized LMO geometry, whereas the radius multiplier s calibrates the magnitude of the completed direction.

Vocabulary geometry. SCION already distinguishes input, hidden, and output layers: for an untied one-hot input it uses a normalized-column oracle (row normalization in our $V \times d$ table orientation) and a Sign output oracle, while under first/last-layer weight sharing it selects Sign for the shared layer (Pethick et al., 2025). Thus SCION Sign is a direct endpoint baseline for our setting. Concurrent symmetry-based work derives equivariant updates for separate input embeddings and *untied* LM heads, including row-normalized, one-sided spectral, hybrid, and centered maps; its reported end-to-end stacks retain AdamW for scalar and vector blocks (Lau & Su, 2026). These results strongly support functional-role-specific vocabulary geometry, but they do not prescribe a joint oracle for one aliased embedding/head tensor. We do not claim novelty for Sign or row-RMS themselves. Our contribution is the exact LMO of a proposed intersection of max-row-ℓ_2 and max-entry balls for the merged tied-table momentum. It contains SCION Sign and row-RMS as endpoints and exposes finite-cap directions between them without adding two separately scaled updates.

Weight tying and mixed gradients. Input–output weight tying was proposed independently by Press & Wolf (2017) and Inan et al. (2016) to reduce the vocabulary-dependent parameter count and often improve language modeling. The update analysis of Press & Wolf (2017) reveals the asymmetry central to our problem: the lookup path contributes only to rows corresponding to observed input tokens, whereas the output-softmax path generally contributes to every vocabulary row. Consequently, a tied table evolves more like the output embedding of an untied model than its input embedding. Later work further distinguishes the structures learned by input and output embeddings (Bertolotti & Cazzola, 2024). Most recently, Lopardo et al. (2026) give causal evidence that output-softmax gradients dominate the tied table early in training, biasing the shared matrix toward the output (unembedding) space, and mitigate this representational bias by rescaling the input gradient. This independently corroborates the sparse-lookup/dense-classifier asymmetry central to our setting; AF-Muon addresses the same imbalance from a complementary direction, changing the LMO geometry used to update the aliased tensor rather than reweighting its merged gradient. Together, these studies motivate treating a tied table as neither a pure lookup table nor a pure output classifier.

Prior work has also addressed conflicts in shared embeddings by changing the parameterization or gradient flow. Projected and generalized tying relax hard equality between input and output representations (Gulordava et al., 2018; Pappas et al., 2018); in the different setting of generator/discriminator embedding sharing, DeBERTaV3 disentangles gradients that otherwise produce "tug-of-war" dynamics (He et al., 2021). AF-Muon instead preserves standard causal-LM weight tying, keeps the single autograd-merged lookup and softmax gradient and one momentum buffer, and changes only the geometry used to update that aliased tensor.

Muon variants. MuonAll pursues an AdamW-free recipe by applying Muon to existing two-dimensional parameters, including embedding and head matrices, and by diagonalizing one-dimensional tensors before Newton–Schulz; its experiments focus on supervised fine-tuning rather than from-scratch pretraining (Page et al., 2025). Very recent work on Muon representation geometry systematically studies how native, nearest-square, skinny, and vector representations alter the polar update (Chang et al., 2026). Other Muon variants combine orthogonalization with element-wise or neuron-wise second-moment adaptation (Si et al., 2025; Li et al., 2025). These approaches either extend a matrix-orthogonalization template to additional stored shapes or restore adaptive second-moment statistics. AF-Muon takes a complementary route: it reserves spectral geometry for hidden linear maps, assigns RMS geometry to one-dimensional auxiliary tensors, and introduces a native finite-cap geometry for the tied vocabulary table, without second-moment state in any branch.

Memory-efficient optimization. A complementary line of work targets optimizer *memory* rather than update geometry. Adafactor factorizes the second moment to sublinear state (Shazeer & Stern, 2018), Adam-mini replaces per-coordinate second moments with a single rate per parameter block (Zhang et al., 2025b), and GaLore maintains moments in a low-rank gradient subspace (Zhao

et al., 2024); minimalist optimizer designs pretrain with substantially reduced persistent state (Glentis et al., 2025). Muon-family methods are already lighter than AdamW because they store one momentum buffer and no second moment (Jordan et al., 2024; Liu et al., 2025). Several recent variants, however, reintroduce second-moment statistics on the orthogonalized update to improve per-neuron conditioning (Si et al., 2025; Li et al., 2025), trading memory back for adaptivity. AF-Muon moves in the opposite direction: it eliminates the auxiliary AdamW second moment entirely and keeps a single buffer per tensor in every branch, so the optimizer is second-moment-free throughout while still assigning role-specific geometry to hidden matrices, the tied table, and vector parameters.

- ❒ Appendix B fixes notation and recalls the standard norm-ball LMOs used as building blocks.
- ❒ Appendix C develops the alias-pullback framework in general: the pullback norm and its intersection ball (Theorem 1), its dual as an infimal convolution (Theorem 2), and the intersection form of simultaneous role-wise redundancy (Theorem 3).
- ❒ Appendix D instantiates it for a hard-tied table with two roles, giving the tied norm (10).
- ❒ Appendix E solves the resulting LMO exactly (Theorem 4).
- ❒ Appendix F computes the dual norm, the objective-retention ratio, and the concentration bound.
- ❒ Appendix G determines the exact symmetry class of the oracle and shows that its departure from full orthogonal equivariance is structural.
- ❒ Appendix H collects the conditional convergence statements and delimits their scope.
- ❒ Appendix I extends the results to three-way tying.

B SETTING, NOTATION, AND STANDARD LMO GEOMETRY

For Appendices B through H, we focus exclusively on decoder-only models; in Appendix I, we demonstrate how our results extend to other architectures.

We consider a decoder-only language model with stored parameters

$$\theta = (W_1, \dots, W_L, E, v_1, \dots, v_K),$$

where $W_\ell \in \mathbb{R}^{m_l \times n_l}$ are hidden Matrix weights, $E \in \mathbb{R}^{V \times d}$ is a hard-tied token-embedding/LM-head table, and v_k are scalar or one-dimensional auxiliary parameters such as normalization gains and biases. Here V is the vocabulary size and d is the hidden width. In stored coordinates, the table serves two roles:

$$\begin{aligned} i &\longmapsto E^\top e_i \in \mathbb{R}^d \qquad \text{(token lookup)} \\ h &\longmapsto Eh \in \mathbb{R}^V \qquad \text{(output classification)}, \end{aligned}$$

where e_i is a vocabulary basis vector, and both roles refer to the same parameter object.
Hard tying requires a single update for the stored table. AF-Muon chooses its update geometry by imposing both a lookup-side row constraint and an output-side coordinate constraint on that update. These constraints are a modeling choice; weight tying alone does not uniquely determine their norms, radii, or learning rates. Weight tying and the distinction between input and output embeddings predate this construction (Press & Wolf (2017); Inan et al. (2016)).

B.1 NOTATIONS

All spaces are finite-dimensional and real. We use $\langle x, y\rangle = x^\top y$ for vectors and $\langle A, B\rangle = \mathrm{tr}(A^T B) = \sum_{ij} A_{ij} B_{ij}$ for matrices. Our norm conventions are

$$\begin{aligned}
&\|A\|_F = (\textstyle\sum_{ij} A_{ij}^2)^{1/2}, && \text{frobenius norm},\\
&\|A\|_{\mathrm{op}} = \sigma_{\max}(A), && \text{spectral/operator norm},\\
&\|A\|_* = \textstyle\sum_j \sigma_j(A), && \text{nuclear norm},\\
&\|A\|_\infty = \max_{ij} |A_{ij}|, && \text{entrywise infinity norm},\\
&\|A\|_1 = \textstyle\sum_{ij} |A_{ij}|, && \text{entrywise one-norm},\\
&\mathrm{RMS}(x) = \|x\|_2 / \sqrt{p} \quad (x \in \mathbb{R}^p), && \text{root-mean-square vector norm}.
\end{aligned}$$

RMS is a scaled Euclidean vector norm, not a matrix operator norm. For $U \in \mathbb{R}^{V\times d}$, write $u_i \in \mathbb{R}^d$ for the transpose of its i-th row, so $U_{i,:} = u_i^T$. A row-energy bound $\|u_i\|_2^2 \le d$ is equivalent to $\mathrm{RMS}(u_i) \le 1$. For a vector b, let $\mathrm{supp}\, b = \{j : b_j \neq 0\}$; throughout, $\mathrm{sign}(0) = 0$.

For a general norm N, its dual is denoted N^*. The notation $N_{\mathrm{tie},c}$ is identical to $\|\cdot\|_{\mathrm{tie},c}$ in the main text. Subscripts such as $N^*_{\mathrm{tie},c}$ refer to dual norms, whereas $\|A\|_*$ always denotes the nuclear norm.

We also write $[t]_+ = \max\{t, 0\}$ for the positive part, $\mathbf{1}$ for an all-ones vector, $\partial\psi$ for the convex subdifferential of ψ, δ_C for the indicator of a convex set C (zero on C, $+\infty$ elsewhere), $\sigma_C(B) = \sup_{U\in C}\langle B, U\rangle$ for its support function, $\sigma_j(A)$ for the j-th singular value, and $O(d)$ for the orthogonal group.

B.2 DUAL NORMS AND LINEAR MINIMIZATION ORACLES

For a norm N on a finite-dimensional inner-product space, define

$$N^*(Z) = \max_{N(U)\le 1} \langle Z, U\rangle, \qquad D_N(Z) \in \arg\max_{N(U)\le 1} \langle Z, U\rangle.$$

We call D_N the aligned LMO direction; the actual minimization step is $-\eta D_N(Z)$. Compactness of the unit ball ensures attainment. For $\eta > 0$,

$$\langle Z, D_N(Z)\rangle = N^*(Z), \qquad \langle Z, -\eta D_N(Z)\rangle = -\eta N^*(Z) < 0 \quad (Z \neq 0).$$

This identity concerns the oracle input Z. If Z is momentum rather than the current gradient, it does not itself establish descent of the current objective. Also, $D_N(Z) \in \partial N^*(Z)$. Positive rescaling leaves the argmax set unchanged; a scale-independent selection satisfies $D_N(\alpha Z) = D_N(Z)$ for $\alpha > 0$. We choose $D_N(0) = 0$.

The following are standard norm-ball LMOs (Rockafellar (1997) and Boyd & Vandenberghe (2004)), not new ingredients of AF-Muon.

1. Spectral norm for hidden matrices. Let $Z = P_r \Sigma_r Q_r^\top$ be a compact SVD containing exactly the $r = \mathrm{rank}(Z)$ positive singular values. Then

$$\max_{\|U\|_{\mathrm{op}}\le 1} \langle Z, U\rangle = \|Z\|_*, \qquad D_{\mathrm{op}}(Z) = P_r Q_r^\top.$$

For $Z = 0$, this compact polar factor is zero. The upper bound follows from spectral/nuclear duality, and equality follows by substitution. Rank-deficient inputs can have other maximizers, including components on null singular subspaces; the compact polar factor fixes the zero action there. A full thin-SVD completion need not make this same selection. Muon's finite Newton–Schulz implementation approximates this ideal direction and requires separate feasibility and accuracy control (Appendix H).

2. RMS norm for vectors. For $u \in \mathbb{R}^p$, let $N_{\mathrm{RMS}}(u) = \|u\|_2 / \sqrt{p}$. Cauchy–Schwarz gives

$$N^*_{\mathrm{RMS}}(b) = \sqrt{p}\,\|b\|_2, \qquad D_{\mathrm{RMS}}(b) = \begin{cases} b/\,\mathrm{RMS}(b), & b \neq 0,\\ 0, & b = 0. \end{cases}$$

This optimizer direction follows from the chosen norm; it is distinct from the RMSNorm architectural operation of Zhang & Sennrich (2019).

3. Max-entry norm. For matrices U, let $N_\infty(U) = \|U\|_\infty$. The dual norm is the entrywise ℓ_1 norm,

$$N_\infty^*(B) = \|B\|_1 ,$$

and a corresponding maximizing direction is

$$D_\infty(B) = \operatorname{sign}(B),$$

applied entrywise. Each nonzero coordinate is maximized by choosing its sign, while zero coordinates are set to zero by convention. This yields the Sign endpoint.

These maps are steepest directions for different geometries, not interchangeable normalizations of a single norm. Norm-dependent layer assignments are studied by Pethick et al. (2025), Large et al. (2024), and Bernstein & Newhouse (2024; 2025). The construction below combines selected role constraints when several roles share one stored tensor.

C THE ALIAS-PULLBACK FRAMEWORK

In an untied model, the input table and output head can have distinct updates. Under hard tying, autograd accumulates their contributions into one gradient on one parameter. Separate role constraints must therefore act on the same stored perturbation. Pullbacks, intersections, support functions, and infimal convolution are standard convex-analysis constructions (Rockafellar (1997)); here they organize the shared-parameter geometry.

C.1 WHERE DOES THE TERMINOLOGY COME FROM?

A *pullback* transfers a structure on a target space to a source through a map. Here a linear role map $A_r : \mathcal{W} \to \mathcal{X}_r$ transfers a norm constraint from the role space to the stored-parameter space.
An *alias*, in programming, is another reference to the same underlying object. In this setting, the roles alias one stored tensor: changing that tensor changes every role view through its corresponding A_r. The resulting feasible set is an intersection: every role constraint must hold simultaneously. Its dual admits a decomposition into latent role covectors. This decomposition is geometric and need not coincide with the separately accumulated computational-path gradients.

We therefore call the resulting construction the *alias-pullback geometry*: it is the pullback to the shared, aliased parameter space of the geometries imposed by its multiple functional roles. Its unit ball is naturally the intersection of the role-wise feasible sets, so a stored update is feasible only when it is simultaneously feasible in every role. Dually, the merged covector can be represented by an optimal decomposition into latent role covectors, which explains the infimal-convolution expression for the dual norm. Finally, any redundancy of the shared parameter must also be compatible with every role, leading to an intersection of the role-wise redundancy subspaces.

C.2 PULLBACK NORM AND INTERSECTION BALL

Let $R \geq 1$. For each $r \in \{1, \dots, R\}$, let $(\mathcal{X}_r, N_r)$ be a normed space and let $\rho_r > 0$ be the corresponding role radius. Define the scaled product norm

$$N_\oplus(x_1, \dots, x_R) = \max_{1 \leq r \leq R} \frac{N_r(x_r)}{\rho_r}, \qquad x_r \in \mathcal{X}_r.$$

Given linear role maps $A_r : \mathcal{W} \to \mathcal{X}_r$, define the alias seminorm

$$N_{\text{alias}}(U) = \max_r \frac{N_r(A_r U)}{\rho_r}, \qquad U \in \mathcal{W}.$$

Equivalently, with $A = (A_1, \dots, A_R) : \mathcal{W} \to \prod_{r=1}^R \mathcal{X}_r$, we have

$$N_{\text{alias}} = N_\oplus \circ A.$$

Note that N_{alias} is a semi-norm on $\mathcal{W}$; it is a norm precisely when $\bigcap_{r=1}^R \ker A_r = \{0\}$.

Theorem 1 (Alias-pullback norm and intersection ball)**.** *If*

$$\bigcap_{r=1}^{R} \ker A_r = \{0\},$$

then N_{alias} *is a norm and*

$$\mathcal{C}_{\text{alias}} = \{U : N_{\text{alias}}(U) \le 1\} = \bigcap_{r=1}^{R} A_r^{-1}\{X : N_r(X) \le \rho_r\}.$$

Proof. Positive homogeneity and nonnegativity follow immediately from the corresponding properties of the norms N_r. Further,

$$N_{\text{alias}}(U+V) \le \max_r \left[\frac{N_r(A_rU)}{\rho_r} + \frac{N_r(A_rV)}{\rho_r}\right] \le N_{\text{alias}}(U) + N_{\text{alias}}(V).$$

If $N_{\text{alias}}(U) = 0$, then U belongs to every $\ker A_r$, hence $U = 0$. The ball identity follows directly from the maximum. Without the common-kernel assumption, this is a semi-norm, becoming a norm on the quotient by $\bigcap_r \ker A_r$. □

C.3 Dual of the alias geometry

The dual evaluates a covector on the stored space by finding its cheapest representation through role covectors. Adjoint maps are taken with respect to the specified inner products.

Theorem 2 (Alias-aware dual norm)**.** *Let* $\mathcal{W}$ *and* $\mathcal{X}_r$ *be finite-dimensional inner-product spaces,* $A_r : \mathcal{W} \to \mathcal{X}_r$ *linear,* N_r *norms, and* $\rho_r > 0$*. Assume explicitly that* $\bigcap_r \ker A_r = \{0\}$*. Then for every* $B \in \mathcal{W}$*,*

$$N^*_{\text{alias}}(B) = \min_{\sum_{r=1}^R A_r^* Z_r = B} \sum_{r=1}^{R} \rho_r N_r^*(Z_r).$$

The minimum is feasible and attained.

Proof. Let $\mathcal{C}_r = \{U : N_r(A_rU) \le \rho_r\}$. Write δ_C for the convex indicator of C and $\sigma_C(B) = \sup_{U \in C}\langle B, U\rangle$ for its support function. Since

$$\delta_{\mathcal{C}_{\text{alias}}} = \sum_r \delta_{\mathcal{C}_r},$$

conjugacy gives the infimal-convolution formula. Each $\mathcal{C}_r$ contains a neighborhood of zero, even when A_r is not injective: continuity of $N_r \circ A_r$ and $\rho_r > 0$ ensure this. The common-interior qualification therefore gives exactness and attainment at finite values:

$$\sigma_{\mathcal{C}_{\text{alias}}}(B) = \min_{\sum_r B_r = B} \sum_r \sigma_{\mathcal{C}_r}(B_r). \tag{7}$$

For $\mathcal{D}_r = \{X : N_r(X) \le \rho_r\}$, $\delta_{\mathcal{C}_r} = \delta_{\mathcal{D}_r} \circ A_r$. The same continuity qualification at $A_r 0 = 0$ yields

$$\sigma_{\mathcal{C}_r}(B_r) = \inf_{A_r^* Z_r = B_r} \sigma_{\mathcal{D}_r}(Z_r) = \inf_{A_r^* Z_r = B_r} \rho_r N_r^*(Z_r). \tag{8}$$

If the constraint is infeasible, both sides equal $+\infty$, using $\inf \varnothing = +\infty$. Otherwise the infimum is a minimum, since a norm is coercive on the closed affine feasible set. Substitution into (7) gives the stated aggregate constraint.

For completeness, define $AU = (A_1U, \dots, A_RU)$. The common-kernel assumption says A is injective, so its adjoint A^* is surjective in finite dimensions. Thus $\sum_r A_r^* Z_r = B$ is feasible for every B. The objective $\sum_r \rho_r N_r^*(Z_r)$ is coercive on the product space, ensuring attainment. Finally $\sigma_{\mathcal{C}_{\text{alias}}} = N^*_{\text{alias}}$ by definition. □

Semi-norm case. If $K = \bigcap_r \ker A_r \neq \{0\}$, the same support-function identity holds with extended-real values. It is finite precisely for $B \in K^\perp = \text{range}(A^*)$, where it describes the quotient dual. It is not a finite norm on all of $\mathcal{W}$.

The minimizing Z_r represent an optimal geometric decomposition of a merged gradient or momentum. They are not required to equal the actual gradients produced by the individual roles, and the oracle need not form or store them.

C.4 REDUNDANCY UNDER ALIASING

Role-wise invariance must be distinguished from invariance of the composed network. Suppose $K_r \subseteq \mathcal{X}_r$ is a linear subspace of perturbations that leave role r unchanged, with its other arguments held fixed. For exact finite invariance, require this property along every translation in K_r on the domain being considered. If K_r instead denotes a Jacobian kernel, the resulting statement is only infinitesimal and local.

Theorem 3 (Simultaneous role-wise redundancy is an intersection)**.** *For linear role maps A_r, the stored perturbations that are redundant in every role are exactly*

$$K_{\text{alias}} := \{U : A_r U \in K_r \text{ for all } r\} = \bigcap_{r=1}^{R} A_r^{-1}(K_r).$$

Proof. A perturbation is role-wise redundant exactly when each role view lies in that role's redundancy subspace. Membership in all preimages is equivalent to membership in their intersection. □

This is a characterization of *simultaneous role-wise* redundancy. Such redundancy is sufficient for invariance of a network composed from those roles, but it need not exhaust the network's invariances. Downstream maps can discard changes, and effects through several paths can cancel. In particular, $K_{\text{alias}} = \{0\}$ does not prove that the full network has no symmetries, parameter degeneracies, or loss-flat directions.

Quotient geometry requires a further distinction. If the objective is invariant along a fixed subspace K, the norm induced on equivalence classes is

$$\overline{N}([U]) = \inf_{k \in K} N_{\text{alias}}(U + k).$$

Identify the quotient with $K^{\perp}$ using the orthogonal projection $P_{K^{\perp}}$. Its unit ball is the *projection*

$$P_{K^{\perp}} \mathcal{C}_{\text{alias}},$$

not generally the section $\mathcal{C}_{\text{alias}} \cap K^{\perp}$. For $B \in K^{\perp}$, an oracle over the projected ball has the same optimal linear value as the original ball. An oracle restricted to the section is a different, potentially more restrictive choice unless projection is nonexpansive in N_{alias}.

For example, in $\mathbb{R}^2$ let $N(x, y) = \sqrt{x^2 + (y - x)^2}$ and $K = \text{span}(e_2)$. The unit ball projects onto $|x| \leq 1$, whereas its section at $y = 0$ is $|x| \leq 1/\sqrt{2}$. Thus orthogonal restriction can reduce the optimum even for a covector orthogonal to the redundant subspace.

These distinctions matter for an untied softmax head, whose common-logit-shift invariance is studied in symmetry-compatible optimizer design (Lau & Su (2026)). A direction that is redundant for the head alone can change the lookup role of a tied table. Appendix D establishes precisely that role-wise obstruction; it makes no claim to classify every symmetry of a complete transformer.

D HARD-TIED TOKEN TABLES

D.1 GRADIENT STRUCTURE OF THE TIED TABLE

Hard tying makes autograd add all computational-path contributions to the same stored gradient. The following formulas use a summed cross-entropy loss. For a weighted or averaged loss, the corresponding position weights or normalization factors are included in each summand. Lookup and output positions can have different index sets.

Proposition 2 (Sparse lookup and generically dense output contributions)**.** *For lookup positions t, token indices i_t, and upstream embedding gradients $g_{x,t} = \nabla_{x_t} \mathcal{L} \in \mathbb{R}^d$,*

$$G_{\text{in}} = \sum_{t \in \mathcal{T}_{\text{in}}} e_{i_t} g_{x,t}^{\top}.$$

Its nonzero rows are contained in the set of token indices used by the lookup. For output positions with $z_t = Eh_t$, $p_t = \mathrm{softmax}(z_t)$, and one-hot target y_t,

$$G_{\mathrm{out}} = \sum_{t \in \mathcal{T}_{\mathrm{out}}} (p_t - y_t) h_t^\top, \qquad G_E = G_{\mathrm{in}} + G_{\mathrm{out}}.$$

The output formula is the direct head contribution, holding h_t fixed in this partial derivative. Dependence of h_t on earlier uses of E is already represented by the lookup-path gradients.

Proof. At a lookup, $x_t = E^\top e_{i_t}$ and $dx_t = (dE)^\top e_{i_t}$. Therefore

$$d\mathcal{L}_{\mathrm{lookup},t} = g_{x,t}^\top (dE)^\top e_{i_t} = \langle e_{i_t} g_{x,t}^\top, dE \rangle.$$

At the head, $\nabla_{z_t} \ell_t = p_t - y_t$ and $dz_t = (dE)h_t$, so

$$d\mathcal{L}_{\mathrm{head},t} = (p_t - y_t)^\top (dE) h_t = \langle (p_t - y_t) h_t^\top, dE \rangle.$$

Summing the differentials over all uses of the stored parameter proves the identity. □

For $V > 1$, finite logits, and a full softmax, each $0 < p_{t,i} < 1$. Thus a nonzero hidden state yields a nonzero vector in every vocabulary row of an individual head contribution. Summed contributions may cancel; individual hidden coordinates may vanish; masking, alternative losses, and numerical underflow can alter support. Consequently full-softmax training does not guarantee that every coordinate of the merged gradient or momentum is nonzero. The support-aware oracle in Appendix E covers these cases.

D.2 Why the untied-head quotient disappears

For an untied output head, adding $\mathbf{1}a^\top$ adds the common scalar $a^\top h$ to every logit. Softmax probabilities are unchanged. When hidden states range over $\mathbb{R}^d$,

$$K_{\mathrm{out}} = \{\mathbf{1}a^\top : a \in \mathbb{R}^d\} \tag{9}$$

is exactly this head-shift subspace. Restricted hidden-state sets can introduce additional degeneracies. The following result concerns the specified head-shift symmetry and the full lookup map.

Proposition 3 (Hard tying removes simultaneous head-shift redundancy)**.** *Assume the input role map is injective, so $K_{\mathrm{in}} = \{0\}$. The output role, under the softmax loss, has head-shift redundancy: for every $a \in \mathbb{R}^d$, the perturbation $\mathbf{1}a^T$ leaves the output probabilities unchanged to first order, so $\mathbf{1}a^\top \in K_{\mathrm{out}}$. Then*

$$K_{\mathrm{alias}} = K_{\mathrm{in}} \cap K_{\mathrm{out}} = \{0\}.$$

In particular, a nonzero head-shift direction $\mathbf{1}a^\top$ is redundant for the output role alone but not simultaneously for the two roles.

Proof. By Theorem 3, $K_{\mathrm{alias}} = K_{\mathrm{in}} \cap K_{\mathrm{out}}$. Since $K_{\mathrm{in}} = \{0\}$, the intersection is $\{0\}$. Directly, replacing E by $E + \mathbf{1}a^\top$ changes every lookup vector by a, which the input role detects. Hence such a perturbation is not redundant once the two roles are tied. □

Proposition 4 (Row-sum identities)**.** *Under the loss convention of Proposition 2,*

$$\mathbf{1}^\top G_{\mathrm{out}} = 0, \qquad \mathbf{1}^\top G_{\mathrm{in}} = \sum_{t \in \mathcal{T}_{\mathrm{in}}} g_{x,t}^\top.$$

Proof. For every output position $t \in \mathcal{T}_{\mathrm{out}}$, the probability distribution p_t and target distribution y_t both sum to one, so $\mathbf{1}^\top (p_t - y_t) = 1 - 1 = 0$. For every input lookup, the basis vector satisfies $\mathbf{1}^\top e_{i_t} = 1$. Summing these identities over their respective position sets yields the result. □

The second sum need not vanish. Hence, projecting the merged tied gradient with $\Pi_\perp = I_V - \mathbf{1}\mathbf{1}^\top / V$ can discard a legitimate input-role contribution. The same issue applies to accumulated tied momentum. AF-Muon therefore uses the merged, unprojected buffer. This conclusion does not exclude an architecture-specific invariance that must be established separately for the composed network.

D.3 THE TIED INTERSECTION NORM

It is worth stating why both role maps are the identity, given that the two roles act on E differently. The alias framework constrains a *perturbation* $U = dE$ of the stored table, not the forward map. A lookup perturbs the embedding of token i by $U^\top e_i = u_i$, and the head perturbs the logits at hidden state h by Uh; both are linear images of the same U and both views live in the same space of $V \times d$ perturbations. Hence $A_{\text{in}} = A_{\text{out}} = \text{Id}$, and the entire role distinction is carried by the *norms* N_{in} and $N_{\text{out},c}$ rather than by the maps; the common-kernel condition of Theorem 1 is then immediate. Appendix C is nevertheless stated for general A_r, since nontrivial role maps arise elsewhere, for instance a factorized or projected head $U \mapsto UP$ or a partially shared table, and since Theorem 3 is stated through preimages.

AF-Muon chooses

$$N_{\text{in}}(U) = \max_i \frac{\|u_i\|_2}{\sqrt{d}}, \qquad N_{\text{out},c}(U) = \frac{\|U\|_\infty}{c}, \quad 1 \le c < \infty.$$

Each norm bounds the quantity its role feeds downstream. On the lookup side, a feasible perturbation moves every token embedding by at most $\text{RMS}(u_i) \le 1$: a bounded root-mean-square displacement per token, uniform over the vocabulary and independent of which tokens the batch happens to contain. On the output side, the logit perturbation at hidden state h is Uh, and for $U \in \mathcal{C}_c$,

$$\|Uh\|_\infty = \max_i |\langle u_i, h\rangle| \le \min\{\, c\,\|h\|_1\,,\ \sqrt{d}\,\|h\|_2\,\}.$$

The two bounds are not comparable: the cap supplies the tighter one exactly when $\|h\|_1 / \|h\|_2 < \sqrt{d}/c$, that is, when the hidden state is concentrated, which is the regime where a single coordinate of an update row would otherwise dominate a logit direction. Neither constraint implies the other in the range $1 < c < \sqrt{d}$, and c sets their relative strength. Figure 1 shows the resulting feasible sets.

The diagonal alias map $U \mapsto (U, U)$ gives

$$N_{\text{tie},c}(U) = \max\left\{ \max_i \frac{\|u_i\|_2}{\sqrt{d}}, \frac{\|U\|_\infty}{c} \right\}. \tag{10}$$

Proposition 5 (Hard-tied intersection geometry)**.** *The functional in (10) is a norm with unit ball*

$$\mathcal{C}_c = \{U : \|u_i\|_2 \le \sqrt{d}\ \forall i,\ |U_{ij}| \le c\ \forall i, j\}. \tag{11}$$

Proof. Apply Theorem 1 with identity role maps, norms N_{in} and $\|\cdot\|_\infty$, and radii 1 and c. The common kernel is zero. Expanding the two constraints gives (11). □

The first constraint bounds row energy; the second caps coordinate magnitude. At $c = 1$ the row constraint is redundant. For $c \ge \sqrt{d}$ the coordinate constraint is redundant. We use $c = \infty$ as shorthand for omitting the latter, with $N_{\text{tie},\infty} = N_{\text{in}}$. The cap changes shape only in the nontrivial range $1 \le c < \sqrt{d}$; an external positive scalar changes the overall radius.

E EXACT SUPPORT-AWARE CLIPPED-ROW ORACLE

The ball 11 is a Cartesian product across vocabulary rows, so it suffices to solve

$$\max_{u \in \mathbb{R}^d} \langle b, u\rangle \quad \text{subject to} \quad \|u\|_2 \le \sqrt{d},\ \|u\|_\infty \le c. \tag{12}$$

Let $S(b) = \operatorname{supp} b$, $k(b) = |S(b)|$, and $r_c^2(b) = \min\{d, c^2 k(b)\}$. For $b \ne 0$, choose $\tau_c(b) \ge 0$ satisfying

$$\sum_{j=1}^{d} \min\{c^2, \tau_c(b)^2 b_j^2\} = r_c^2(b), \tag{13}$$

and set $\tau_c(0) = 0$. Define

$$[L_c(b)]_j = \operatorname{sign}(b_j) \min\{c, \tau_c(b)|b_j|\}. \tag{14}$$

When $c^2 k(b) \le d$ and $b \neq 0$, every $\tau \ge c/\min_{j \in S(b)} |b_j|$ solves 13 and produces the same direction. When $c^2 k(b) > d$, the root is unique.

definition 1 (Support-preserving maximizer). *A feasible vector u for 12 is called* support-preserving *if* $\operatorname{supp} u = \operatorname{supp} b$. *Equivalently, u places no energy on coordinates where the input vanishes. When $c^2 k(b) < d$, the supported coordinates do not exhaust the Euclidean budget, so any vector supported off* $\operatorname{supp} b$ *with small enough norm may be added without changing the objective or violating feasibility. All maximizers agree on* $\operatorname{supp} b$*; the canonical selection (14) is the one that zeroes the remainder. This convention is what makes L_c single-valued as a map, keeps it odd and positively scale invariant.*

Theorem 4 (Exact support-aware clipped-row LMO). *For every finite $c \ge 1$, $L_c(b)$ maximizes objective 12. It is the unique support-preserving maximizer and satisfies*

$$\operatorname{supp} L_c(b) = \operatorname{supp} b, \qquad \|L_c(b)\|_2^2 = \min\{d, c^2 k(b)\}. \tag{15}$$

The zero input is interpreted as $L_c(0) = 0$. More explicitly,

$$c^2 k(b) \le d \quad \Longrightarrow \quad L_c(b) = c \operatorname{sign}(b), \tag{16}$$

whereas in the radius-limited case,

$$[L_c(b)]_j = \operatorname{sign}(b_j) \min\{c, \tau |b_j|\}, \qquad \sum_j \min\{c^2, \tau^2 b_j^2\} = d. \tag{17}$$

Proof. For $b = 0$, zero is the unique support-preserving feasible vector and is optimal. Let $b \neq 0$. Aligning each supported coordinate with b_j cannot decrease the objective and preserves feasibility. Setting coordinates outside $S(b)$ to zero also preserves feasibility and the objective. Write $q_j = |b_j| > 0$ on $S(b)$ and $u_j = \operatorname{sign}(b_j) a_j$. The supported problem is

$$\max_a \sum_{j \in S(b)} q_j a_j \quad \text{subject to} \quad \sum_{j \in S(b)} a_j^2 \le d, \quad 0 \le a_j \le c. \tag{18}$$

If $c^2 k(b) \le d$, the vector $a_j = c$ is feasible and maximizes every positive term individually, proving equation 16. These supported coordinates are uniquely determined. If $c^2 k(b) < d$ and zero coordinates exist, the unrestricted problem can also have off-support components using the remaining radius; the canonical selection excludes them. If $c^2 k(b) = d$, no off-support energy remains.

If $c^2 k(b) > d$, the Euclidean constraint must be active; otherwise at least one coordinate is below its cap and can be increased slightly to improve the objective while remaining feasible. We establish the unique optimizer for this case next.

Problem 18, with negated objective, is convex and strictly feasible by taking all supported a_j sufficiently small and positive. KKT conditions are therefore necessary and sufficient. With multipliers $\lambda \ge 0$ for $\sum a_j^2 \le d$, $\alpha_j \ge 0$ for $-a_j \le 0$, and $\beta_j \ge 0$ for $a_j - c \le 0$, stationarity is

$$-q_j + 2\lambda a_j - \alpha_j + \beta_j = 0. \tag{19}$$

No optimal a_j can be zero: then $\beta_j = 0$ and 19 would imply $-q_j - \alpha_j = 0$. Thus $a_j > 0$ and $\alpha_j = 0$. Also $\lambda > 0$: if $\lambda = 0$, then $\beta_j = q_j > 0$ forces all $a_j = c$, contradicting $c^2 k(b) > d$.

For $a_j < c$, equation 19 gives $a_j = q_j/(2\lambda)$. For $a_j = c$, it gives $q_j/(2\lambda) \ge c$. Hence

$$a_j = \min\{c, \tau q_j\}, \qquad \tau = (2\lambda)^{-1} > 0. \tag{20}$$

The active radius constraint is 17. The function $\phi(\tau) = \sum_{j \in S(b)} \min\{c^2, \tau^2 q_j^2\}$ is continuous, equals zero at zero, and reaches $c^2 k(b) > d$. It is strictly increasing before all terms saturate: if $0 \le \tau_1 < \tau_2$ and $\phi(\tau_1) < c^2 k(b)$, at least one unsaturated term increases. Thus its level-d root exists and is unique. Equations 19–20 and complementary slackness certify optimality. All supported coordinates are positive, and their squared norm is d; off-support motion is then impossible. This proves 15 and uniqueness. □

Corollary 1 (Full-table LMO). *Set $[D_{\mathrm{tie},c}(B)]_{i,:} = L_c(b_i)^\top$. Then*

$$D_{\mathrm{tie},c}(B) \in \operatorname*{arg\,max}_{N_{\mathrm{tie},c}(U) \le 1} \langle B, U \rangle.$$

Proof. Both feasibility and the objective $\langle B, U\rangle = \sum_i \langle b_i, u_i\rangle$ separate by row. □

Corollary 2 (Endpoints)**.** *With* $\operatorname{sign}(0) = 0$, $L_1(b) = \operatorname{sign}(b)$. *For* $c \geq \sqrt{d}$ *(including the shorthand* $c = \infty$*),*

$$L_c(b) = \begin{cases} \sqrt{d}\, b/\left\|b\right\|_2 = b/\operatorname{RMS}(b), & b \neq 0, \\ 0, & b = 0. \end{cases}$$

Proof. At $c = 1$, the box already implies the row-radius bound. At $c \geq \sqrt{d}$, $\left\|u\right\|_\infty \leq \left\|u\right\|_2 \leq \sqrt{d} \leq c$, so the cap is redundant.
The canonical map is odd and positively scale invariant: $L_c(-b) = -L_c(b)$ and $L_c(\alpha b) = L_c(b)$ for $\alpha > 0$. □

Numerical implementation. Use 16 directly for support-limited rows. Otherwise bracket the unique root of 17, doubling an upper endpoint until $\phi(\tau_{\text{hi}}) \geq d$, and bisect. Returning the lower endpoint preserves feasibility in exact arithmetic. A finite solve is approximate; floating-point overflow, feasibility, and objective error must be controlled rather than inferred from Theorem 4. Appendix H states the general framework into which these errors enter.

The tied update in one display. Collecting the pieces, the stored table is updated from a single merged buffer B_t by

$$E_{t+1} = E_t - \eta_{\text{tie},t}\, D_{\text{tie},c}(B_t), \qquad [D_{\text{tie},c}(B_t)]_{i,:} = L_c(b_{t,i})^\top,$$

with L_c as in (14). There is no projection, no split of B_t into lookup and head parts, and no second buffer; the role structure enters only through the feasible set $\mathcal{C}_c$ over which L_c optimizes. Rows with $b_{t,i} = 0$ receive no update, which is the support-awareness distinguishing L_c from a row-RMS normalization applied after densification.

F Dual geometry, objective retention, and concentration control

We characterize the support function optimized by the exact oracle, its same-input objective relative to row-RMS, and its support-dependent concentration bound. The convex-duality identities used here are classical; the formulas below specialize them to the chosen tied-table geometry.

F.1 The tied dual norm as an infimal convolution

The row norm N_{in} has dual $N_{\text{in}}^*(X) = \sqrt{d}\sum_{i=1}^V \left\|x_i\right\|_2$, since its ball is a product of radius-$\sqrt{d}$ Euclidean balls. The dual of $\left\|\cdot\right\|_\infty / c$ is $c\left\|\cdot\right\|_1$.

Theorem 5 (Tied dual as infimal convolution)**.** *For* $B \in \mathbb{R}^{V\times d}$ *and finite* $c \geq 1$*,*

$$N_{\text{tie},c}^*(B) = \min_{X+Y=B} \left[\sqrt{d}\sum_{i=1}^V \left\|x_i\right\|_2 + c\left\|Y\right\|_1\right]. \tag{21}$$

Proof. Apply Theorem 2 with identity role maps, norms N_{in} and $\left\|\cdot\right\|_\infty$, and radii 1 and c. Its constraint is $X + Y = B$, and attainment follows from that theorem. □

The tied geometry prices the merged momentum by its cheapest split into a row-structured component X (group-$\ell_{2,1}$ cost) and an entrywise-spiky component Y (ℓ_1 cost). This is an implicit geometric decomposition: AF-Muon never stores or solves for X, Y, and does not claim they equal the actual lookup and softmax gradients; the primal clipped-row oracle attains the same optimal value with one merged buffer. The infimal-convolution identity itself is classical (Rockafellar (1997)); the novelty is its role as the exact dual of the tied-table geometry.

F.2 Explicit row support function and objective retention

Define the row support function

$$h_c(b) := \max\{\langle b, u\rangle : \|u\|_2 \le \sqrt{d}, \|u\|_\infty \le c\};$$

by row-Cartesian structure,

$$N^*_{\mathrm{tie},c}(B) = \sum_{i=1}^{V} h_c(b_i).$$

With the capped active set $A = \{j : \tau_c(b)|b_j| \ge c\}$, Theorem 4 evaluates it in closed form: in a radius-limited row,

$$\tau_c(b) = \sqrt{\frac{r_c^2(b) - c^2|A|}{\sum_{j\notin A} b_j^2}}, \qquad h_c(b) = c\sum_{j\in A}|b_j| + \tau_c(b)\sum_{j\notin A} b_j^2;$$

in the support-limited case $c^2k(b) \le d$, $h_c(b) = c\,\|b\|_1$. At the endpoints, $h_1(b) = \|b\|_1$ and $h_\infty(b) = \sqrt{d}\,\|b\|_2$, matching the Sign and row-RMS duals.

Determining the active set. The pair $(A, \tau_c(b))$ above is characterized jointly, not sequentially: A is defined through $\tau_c(b)$ and $\tau_c(b)$ through A. In a radius-limited row, they can be identified in closed form by sorting. Since $a_j = \min\{c, \tau q_j\}$ in (20) saturates exactly when $q_j \ge c/\tau$, the capped coordinates are the ones of largest magnitude. Order the support so that $|b_{(1)}| \ge \cdots \ge |b_{(k)}|$, so that $A = \{(1), \ldots, (m)\}$ for some $0 \le m < k$, and for each candidate m put

$$\tau^{(m)} = \sqrt{\frac{d - c^2 m}{\sum_{j>m} b_{(j)}^2}}.$$

Accept the unique m satisfying $\tau^{(m)}|b_{(m)}| \ge c > \tau^{(m)}|b_{(m+1)}|$, with the conventions $|b_{(0)}| = +\infty$ and $|b_{(k+1)}| = 0$. This is an exact $O(d \log d)$ alternative to the bisection of Appendix E; bisection is preferred in the implementation because it batches across rows and avoids the sort.

Proposition 6 (Objective retention is a dual-norm ratio)**.** *For $B \neq 0$, the same-input row-RMS retention is*

$$\mathcal{R}_c(B) := \frac{\langle B, D_{\mathrm{tie},c}(B)\rangle}{\langle B, D_{\mathrm{tie},\infty}(B)\rangle} = \frac{N^*_{\mathrm{tie},c}(B)}{\sqrt{d}\sum_i \|b_i\|_2}.$$

Proof. The numerator is the dual norm by the exact LMO identity:

$$\langle B, D_{\mathrm{tie},c}(B)\rangle = N^*_{\mathrm{tie},c}(B).$$

At $c = \infty$ the rows decouple into radius-$\sqrt{d}$ Euclidean balls, so

$$\langle B, D_{\mathrm{tie},\infty}(B)\rangle = \sqrt{d}\sum_i \|b_i\|_2.$$

□

If $c_1 \le c_2$, then $\mathcal{C}_{c_1} \subseteq \mathcal{C}_{c_2}$, so retention is nondecreasing in c. More precisely,

$$\frac{1}{\sqrt{d}} \le \mathcal{R}_1(B) = \frac{\|B\|_1}{\sqrt{d}\sum_i \|b_i\|_2} \le \mathcal{R}_c(B) \le 1.$$

These are fixed-input geometric comparisons, not predictions of future training loss.

F.3 SUPPORT-AWARE CONCENTRATION (PAPR) BOUNDS

For a nonzero row direction, define

$$\mathrm{PAPR}(u) := \frac{d\,\|u\|_\infty^2}{\|u\|_2^2}.$$

This ratio is unchanged by nonzero scalar multiplication. It always lies in $[1, d]$; if u has k nonzero entries, it is at least d/k.

Intuition Behind it. The acronym abbreviates *peak-to-average power ratio*, borrowed from signal processing: reading u as a length-d signal, $\|u\|_2^2/d$ is its mean squared coordinate (average power) and $\|u\|_\infty^2$ its largest (peak), so $\mathrm{PAPR}(u)$ measures how far one coordinate dominates the row. A flat row has $\mathrm{PAPR} = 1$; a row whose energy sits in a single coordinate has $\mathrm{PAPR} = d$. This is the quantity the cap is designed to control, since a high-PAPR update row displaces one embedding coordinate, and correspondingly one logit direction, far more than the row's overall size suggests.

Proposition 7 (Support-aware PAPR bound)**.** *For finite $c \geq 1$ and $u = L_c(b) \neq 0$,*

$$\mathrm{PAPR}(u) \leq \frac{dc^2}{\min\{d, c^2 k(b)\}}. \tag{22}$$

In particular,

$$c^2 k(b) \geq d \quad \Longrightarrow \quad \mathrm{PAPR}(u) \leq c^2, \tag{23}$$

where the universal upper bound d can further sharpen this to $\min\{c^2, d\}$. In the support-limited case,

$$c^2 k(b) \leq d \quad \Longrightarrow \quad \mathrm{PAPR}(u) = \frac{d}{k(b)}. \tag{24}$$

At $c^2 k(b) = d$, the formulas coincide.

Proof. The coordinate constraint gives $\|u\|_\infty^2 \leq c^2$ and Theorem 4 gives $\|u\|_2^2 = \min\{d, c^2 k(b)\}$. Substitution proves 22–23. In the support-limited case, precisely $k(b)$ entries have magnitude c, giving 24. The general lower bound follows from $\|u\|_2^2 \leq k\,\|u\|_\infty^2$. □

The support condition in 22 is essential. For example, with $d = 16$, $c = 2$, and $b = e_1$, the canonical update is $2e_1$ and its PAPR is 16, not at most $c^2 = 4$. The cap controls absolute coordinate size but cannot create additional supported coordinates. At $c = 1$, a dense Sign row has PAPR one; a sparse Sign row has PAPR $d/k(b)$. Row-RMS has

$$\mathrm{PAPR}(L_\infty(b)) = d\frac{\|b\|_\infty^2}{\|b\|_2^2},$$

which reaches d for a one-coordinate input and approaches d for dense inputs dominated by one coordinate.

Shape versus scale. For a positive scalar s,

$$s\mathcal{C}_c = \{U : \|u_i\|_2 \leq s\sqrt{d}\,\forall i,\ |U_{ij}| \leq sc\,\forall i, j\}.$$

Scaling the completed direction therefore rescales the row radius and cap together, preserving their ratio and PAPR. For a support-sufficient row the scaled energy is $s^2 d$; for a support-limited row it is $s^2 c^2 k(b)$. A uniform scale does not force all nonzero rows to have equal energy when their supports differ.

AF-Muon applies s once through the tied learning rate in main-text equation 6,

$$\eta_{\mathrm{tie},t} = \eta_{M,t}\frac{\rho_{\mathrm{out}}}{\rho_{\mathrm{hid}}}\frac{s}{d}.$$

The width factor and radius ratio are design choices motivated by norm-based scaling and evaluated empirically. Neither their numerical values nor width-independent convergence constants follow from the alias-pullback theorem.

G SYMMETRY AND ITS LIMITATION

The oracle's symmetry class both certifies its invariances and delimits its relationship to fully rotation-equivariant designs such as the LPRO class of Lau & Su (2026). The statements here concern equivariance of the oracle as a map on its input. They do not establish that the full network, loss, data, or numerical implementation has the same symmetry. We prove the positive equivariance result, record the exact range of the negative result, and then show that the negative result is structural, not accidental.

Theorem 6 (Signed-permutation equivariance)**.** *Let $P \in \mathbb{R}^{V\times V}$ be a vocabulary permutation and $Q \in \mathbb{R}^{d\times d}$ a signed permutation (hyperoctahedral group H_d). Then*

$$D_{\mathrm{tie},c}(PBQ^\top) = P\, D_{\mathrm{tie},c}(B)\, Q^\top.$$

Proof. Row permutations permute the independent row problems. Signed feature permutations preserve Euclidean norms and permute absolute coordinate values, hence preserve the max-entry norm and therefore the feasible set and the Frobenius pairing. They also transport the support set, preserve its size, and leave the radius/root equation unchanged. Thus the canonical support-preserving selection, including zero rows and the support-limited case, commutes with the transformation; sign is odd. The displayed equality follows. □

Exact range of the negative result. If $1 \le c < \sqrt{d}$, the oracle is not equivariant under all feature rotations. Because $D_{\mathrm{tie},c}$ acts row by row, it suffices to exhibit the failure for the row map L_c; the matrix statement follows by placing the vector in one row of B and zeroing the rest. Put $q = \mathbf{1}/\sqrt{d}$ and choose $R \in O(d)$ with $Re_1 = q$. The input e_1 has $k = 1$, so $c^2 k \le d$ and (16) applies; the input q has $k = d$, so $c^2 k \ge d$ and the row is radius-limited, with uniform entries forcing $\tau_c(q)|q_j| = 1 \le c$ for every j. Hence

$$\begin{gathered} L_c(e_1) = c\, e_1, \quad L_c(q) = \mathbf{1} = \sqrt{d}\, q \\ \Downarrow \\ R\, L_c(e_1) \neq L_c(Re_1) \quad (c\, q \neq \sqrt{d}\, q), \end{gathered}$$

Note where equivariance fails: the two outputs are parallel but carry different energies, since a support-limited row is allowed only c^2 units of squared norm while its dense rotation receives the full budget d. For every $c \ge \sqrt{d}$, including finite caps, the constraint is redundant, and the oracle equals row-RMS, so full right-$O(d)$ equivariance holds. When $d = 1$, there is no nontrivial cap range. Thus only the active-cap range is excluded from the left-permutation/right-orthogonal (LPRO) class of Lau & Su (2026). The next theorem shows this exclusion is structural, not an implementation artifact.

Theorem 7 (No-go: uniqueness of row-RMS under full orthogonal equivariance)**.** *Let $\varphi : \mathbb{R}^d \setminus \{0\} \to \mathbb{R}^d$ satisfy*

1. *right-orthogonal equivariance: $\varphi(Rb) = R\,\varphi(b)$ for every $R \in O(d)$;*
2. *fixed row energy: $\|\varphi(b)\|_2 = \sqrt{d}$;*
3. *positive alignment: $\langle b, \varphi(b)\rangle > 0$.*

Then necessarily

$$\varphi(b) = \sqrt{d}\,\frac{b}{\|b\|_2},$$

i.e. φ is row-RMS. Positive scale invariance follows from this conclusion and need not be assumed separately.

Proof. Fix $b \neq 0$. If $R \in O(d)$ fixes b, equivariance gives

$$\varphi(b) = \varphi(Rb) = R\varphi(b),$$

so every orthogonal map fixing b also fixes $\varphi(b)$. For $d \ge 2$, a reflection fixing b and reversing a vector in $b^\perp$ shows that $\varphi(b)$ has no component in $b^\perp$; for $d = 1$, collinearity is immediate. Thus $\varphi(b) = \alpha(b)b$. Fixed norm gives $|\alpha(b)| = \sqrt{d}/\|b\|_2$, and positive alignment selects the positive sign. The positive-scale-invariance statement follows immediately from the displayed formula. □

For a row-separable rule, these conditions uniquely select row-RMS. They do not imply that dropping row separability is the only alternative, nor that signed permutations form a maximal symmetry group for every conceivable concentration-control method. In fact, even without fixed energy, any nonzero positively aligned orthogonally equivariant row map is parallel to b, so its PAPR is

$$d\,\frac{\|b\|_\infty^2}{\|b\|_2^2},$$

and cannot have a uniform ceiling below d.

AF-Muon chooses coordinate-dependent control in the active-cap range. Orthogonally invariant constraints, such as an additional spectral constraint, describe different geometries and can couple rows; Theorem 7 does not characterize such coupled algorithms. Support-limited clipped rows also need not have fixed energy, so Theorem 7 alone is not a proof of their failure of equivariance; the explicit counterexample above is.

Relatedly, the "Sign as spectral" representation of Lau & Su (2026) embeds the vectorized matrix in a large diagonal matrix and preserves only coordinate geometry; AF-Muon's finite cap remains coordinate-aware and is not claimed to convert Sign into a fully spectral or LPRO update.

H Exact oracle properties and conditional convergence bounds

The theory in this appendix characterizes the optimization geometry of AF-Muon and gives conditional descent bounds for idealized variants. We distinguish three regimes:

1. Exact deterministic oracle: current full gradients, exact LMOs, and exact arithmetic.
2. Inexact oracle: momentum estimation, approximate polar maps, numerical bisection, and stabilization are represented by explicit estimator and oracle-error terms.
3. Practical stochastic training: no unconditional convergence theorem is claimed. A theorem for the stochastic implementation would require additional assumptions controlling stochastic estimator error, momentum lag, oracle error, feasibility violation, and their interaction with the learning-rate schedule.

Thus the results below are exact oracle properties and conditional convergence bounds for idealized variants, not a convergence guarantee for the full implementation in Algorithm 1. This scope mirrors the analysis frontier for LMO-based LLM optimizers, where conditional convergence theory is developed by Riabinin et al. (2025) and the stochastic/momentum analysis of nonlinear optimizer maps is left open by Lau & Su (2026). All bounds below concern the stated fixed geometry and smooth objective, and they do not assert dimension-independent constants. Practical stochastic pretraining requires quantitative estimator, oracle, and schedule assumptions; unbiased stochastic gradients alone are insufficient.

H.1 Product geometry

Partition parameters into blocks $x = (x_1, \dots, x_B)$, block b with norm N_b and radius $r_b > 0$. The radii absorb branch learning-rate ratios, tied scale/width factors, and shape factors. Define

$$N_{\mathcal{X}}(s) = \max_b \frac{N_b(s_b)}{r_b}.$$

Then, by independent block-wise maximization,

$$N_{\mathcal{X},*}(g) = \sum_b r_b N_{b,*}(g_b), \qquad D_{\mathcal{X}}(g)_b = r_b D_{N_b}(g_b).$$

Idealized role-specific AF-Muon is thus one LMO of a weighted product norm whenever all branches share the scalar schedule a_t, the same product-norm viewpoint as modular-norm methods Large et al. (2024); Pethick et al. (2025). Time-varying relative branch rates generally require a time-varying-geometry analysis.

H.2 Exact deterministic descent

Assume $f \geq f_{\inf} > -\infty$, $L > 0$, and, on the region used by the iterates,

$$f(x+s) \leq f(x) + \langle \nabla f(x), s \rangle + \frac{L}{2} N_{\mathcal{X}}(s)^2.$$

For $x_{t+1} = x_t - \eta_t D_{\mathcal{X}}(\nabla f(x_t))$, $\eta_t \geq 0$, and $\sum_{t<T} \eta_t > 0$, the exact LMO identity and feasibility imply

$$\min_{t<T} N_{\mathcal{X},*}(\nabla f(x_t)) \leq \frac{\Delta + \frac{L}{2} \sum_{t<T} \eta_t^2}{\sum_{t<T} \eta_t}, \qquad \Delta = f(x_0) - f_{\inf}.$$

Indeed, the one-step decrease is at least $\eta_t N_{\mathcal{X},*}(\nabla f(x_t)) - L\eta_t^2/2$; summing proves the bound. If $\Delta > 0$, the horizon-dependent constant step $\eta = \sqrt{2\Delta/(LT)}$ gives $\sqrt{2L\Delta/T}$. For diminishing steps,

$$\sum_t \eta_t = \infty, \quad \sum_t \eta_t^2 < \infty \quad \Longrightarrow \quad \liminf_t N_{\mathcal{X},*}(\nabla f(x_t)) = 0.$$

Under the additional norm-PL condition

$$N_{\mathcal{X},*}(\nabla f(x))^2 \geq 2\mu(f(x) - f_*), \qquad 0 < \mu \leq L,$$

the gradient-dependent step $\eta_t = N_{\mathcal{X},*}(\nabla f(x_t))/L$ gives

$$f(x_{t+1}) - f_* \leq (1 - \mu/L)(f(x_t) - f_*).$$

This is a different schedule from practical AF-Muon. A generic finite-dimensional smoothness constant need not be width- or vocabulary-independent, so these bounds do not by themselves prove hyperparameter transfer. This is the standard normalized-steepest-descent argument Pethick et al. (2025); Riabinin et al. (2025).

H.3 Master inexact-oracle theorem

Let $g_t = \nabla f(x_t)$, let z_t be the implemented oracle input on the same gradient scale (e.g. a momentum estimate), and let $e_t = z_t - g_t$. Assume the implemented direction $\widehat{D}_t = \widehat{D}(z_t)$ satisfies

$$N_{\mathcal{X}}(\widehat{D}_t) \leq 1, \qquad \langle z_t, \widehat{D}_t \rangle \geq N_{\mathcal{X},*}(z_t) - \delta_t, \quad \delta_t \geq 0,$$

and update $x_{t+1} = x_t - \eta_t \widehat{D}_t$. This is the bound through which all AF-Muon approximations enter: momentum lag, stochastic estimation, Newton–Schulz polar error, tied bisection error, and stabilization.

Theorem 8 (Master inexact-oracle bound)**.** *Under L-smoothness in $N_{\mathcal{X}}$,*

$$f(x_{t+1}) \leq f(x_t) - \eta_t N_{\mathcal{X},*}(g_t) + 2\eta_t N_{\mathcal{X},*}(e_t) + \eta_t \delta_t + \frac{L}{2}\eta_t^2,$$

and consequently

$$\frac{\sum_{t<T} \eta_t N_{\mathcal{X},*}(g_t)}{\sum_{t<T} \eta_t} \leq \frac{\Delta + 2\sum_{t<T} \eta_t N_{\mathcal{X},*}(e_t) + \sum_{t<T} \eta_t \delta_t + \frac{L}{2} \sum_{t<T} \eta_t^2}{\sum_{t<T} \eta_t}.$$

Proof. Duality and feasibility give

$$\langle e_t, \widehat{D}_t \rangle \leq N_{\mathcal{X},*}(e_t),$$

so

$$\langle g_t, \widehat{D}_t \rangle = \langle z_t, \widehat{D}_t \rangle - \langle e_t, \widehat{D}_t \rangle \geq N_{\mathcal{X},*}(z_t) - \delta_t - N_{\mathcal{X},*}(e_t).$$

The reverse triangle inequality gives $N_{\mathcal{X},*}(z_t) \geq N_{\mathcal{X},*}(g_t) - N_{\mathcal{X},*}(e_t)$. Combining,

$$\langle g_t, \widehat{D}_t \rangle \geq N_{\mathcal{X},*}(g_t) - 2N_{\mathcal{X},*}(e_t) - \delta_t.$$

Insert this into the smoothness inequality with $s = -\eta_t \widehat{D}_t$ and telescope. □

For this bound to tend to zero, it suffices that $\sum_{t<T} \eta_t \to \infty$, $\sum_{t<T} \eta_t^2 / \sum_{t<T} \eta_t \to 0$, and both weighted-average error terms vanish. These are sufficient conditions for the bound, not necessary conditions for actual convergence. A persistent error bound yields a residual upper bound; it does not by itself prove that the algorithm must have a nonzero error floor.

Feasibility inflation. If instead $N_{\mathcal{X}}(\widehat{D}_t) \le 1 + \kappa_t$, $\kappa_t \ge 0$, with the same objective-accuracy condition, replace $2N_{\mathcal{X},*}(e_t)$ by $(2+\kappa_t)N_{\mathcal{X},*}(e_t)$ and the curvature term by $L\eta_t^2(1+\kappa_t)^2/2$. A five-step bfloat16 Newton–Schulz map is not automatically covered by a uniform small δ_t or κ_t; those bounds require evidence for the actual implementation.

Momentum and stabilization. For $B_t = \mu B_{t-1} + g_t$, $B_{-1} = 0$, and $0 \le \mu < 1$, define $S_t = \sum_{j=0}^{t} \mu^{t-j}$ and $z_t = B_t/S_t$. The exact scale-invariant oracle satisfies $D(B_t) = D(z_t)$, and

$$N_{\mathcal{X},*}(z_t - g_t) \le \sum_{j=0}^{t} \frac{\mu^{t-j}}{S_t} N_{\mathcal{X},*}(g_j - g_t).$$

If $N_{\mathcal{X},*}(g_j - g_t) \le L_g N_{\mathcal{X}}(x_j - x_t)$ and $N_{\mathcal{X}}(x_{k+1} - x_k) \le h$, this is at most $L_g h\mu/(1-\mu)$. The Lipschitz-gradient assumption is explicit; the one-sided descent inequality alone is not used to assert it. Zero-initialized Nesterov-style combinations can likewise be normalized by their total nonnegative weight. A stabilized implementation may lose exact scale invariance: normalize only in the analysis of its actual direction and measure δ_t against that z_t. For example,

$$\widehat{D} = b/\max\{\mathrm{RMS}(b), \varepsilon\}$$

is RMS-feasible with exact deficit

$$\sqrt{p}\|b\|_2 - \frac{\|b\|_2^2}{\max\{\mathrm{RMS}(b), \varepsilon\}}$$

relative to input b.

For stochastic gradients the bound holds pathwise whenever its assumptions hold; expectations require integrable error bounds. In general

$$\mathbb{E}[D(G_t) \mid \mathcal{F}_t] \ne D(g_t)$$

even if $\mathbb{E}[G_t \mid \mathcal{F}_t] = g_t$. Thus unbiased gradients do not give unbiased normalized directions, and with fixed batch and momentum the estimation error generally does not vanish.

H.4 Decoupled decay as conditional gradient

The decayed hidden-matrix update

$$x^+ = (1 - \eta\lambda)x - \eta r D_N(g)$$

is not gradient descent on $f(x) + \frac{\lambda}{2}\|x\|_2^2$; the LMO is applied to g, not to $g + \lambda x$. It has an exact conditional-gradient reading instead.

Theorem 9 (Decoupled decay as a Frank–Wolfe step)**.** *Let $r > 0$, $\lambda > 0$, $R = r/\lambda$, $C = \{x : N(x) \le R\}$, $g = \nabla f(x)$, and $\gamma = \eta\lambda \in [0,1]$. With $v(g) = -RD_N(g) \in \arg\min_{v \in C}\langle g, v\rangle$,*

$$x^+ = (1-\eta\lambda)x - \eta r D_N(g) = (1-\gamma)x + \gamma v(g).$$

If $x \in C$, this is a feasible Frank–Wolfe step. If $x \notin C$, it is the same algebraic update but is not a feasible Frank–Wolfe iterate. Convexity preserves feasibility from $x_0 \in C$. Also

$$[N(x_{t+1}) - R]_+ \le (1-\gamma_t)[N(x_t) - R]_+.$$

Thus distance in norm to the radius bound tends to zero if $\prod_t(1-\gamma_t) = 0$, for example if $\sum_t \gamma_t = \infty$. Positive but summable steps alone do not ensure attraction.

Proof. The scaled negative oracle minimizes $\langle g, v\rangle$ over C; substitution proves the identity. Convexity preserves feasibility from $x_0 \in C$. For the norm recursion, $N(v(g)) \le R$ and $\gamma_t \in [0,1]$ give $N(x_{t+1}) \le (1-\gamma_t)N(x_t) + \gamma_t R$, hence $N(x_{t+1}) - R \le (1-\gamma_t)\big(N(x_t) - R\big)$; taking positive parts and using monotonicity of $[\cdot]_+$ together with $1 - \gamma_t \ge 0$ yields the stated inequality. □

For feasible iterates, the Frank–Wolfe gap

$$G_C(x) = \langle \nabla f(x), x - v(\nabla f(x))\rangle \ge 0.$$

If f is L-smooth in N on C, let $D_C = \sup_{x,y\in C} N(x-y) \le 2R$. Smoothness gives

$$f(x_{t+1}) \le f(x_t) - \gamma_t G_C(x_t) + \frac{L}{2} D_C^2 \gamma_t^2,$$

and

$$\min_{t<T} G_C(x_t) \le \frac{f(x_0) - f_{\inf} + \frac{L}{2} D_C^2 \sum_{t<T} \gamma_t^2}{\sum_{t<T} \gamma_t}.$$

The denominator must be positive. This stationarity bound follows directly by telescoping and does not require convexity of f. For convex f, it also controls suboptimality through $f(x) - \min_C f \le G_C(x)$; classical convex Frank–Wolfe analysis is discussed by Jaggi (2013). For a decayed block of a coupled network, this isolated-block bound cannot be applied while ignoring other simultaneous block updates; the following theorem handles them jointly.

H.5 Hybrid stationarity

Theorem 10 (Hybrid stationarity)**.** *Let $f \ge f_{\inf}$ satisfy the joint bound*

$$f(x+s) \le f(x) + \sum_b \langle g_b, s_b\rangle + \frac{1}{2}\sum_b L_b N_b(s_b)^2.$$

Partition stored blocks into decayed blocks $\mathcal{H}$ and nondecayed blocks $\mathcal{A}$ (including the tied table). Take fixed $r_b, \bar\eta_b > 0$, $\lambda_h > 0$, schedules $\eta_{b,t} = a_t\bar\eta_b$, $a_t \ge 0$, exact current gradients and LMOs, and $a_t\bar\eta_h\lambda_h \le 1$. Assume $x_{h,0} \in C_h = \{x_h : N_h(x_h) \le r_h/\lambda_h\}$. Put

$$G_h(x) = \langle g_h, x_h + (r_h/\lambda_h) D_{N_h}(g_h)\rangle,$$
$$S(x) = \sum_{h\in\mathcal{H}} \bar\eta_h\lambda_h G_h(x) + \sum_{a\in\mathcal{A}} \bar\eta_a r_a N_a^*(g_a),$$

and

$$C_0 = \sum_{h\in\mathcal{H}} L_h(2\bar\eta_h r_h)^2 + \sum_{a\in\mathcal{A}} L_a(\bar\eta_a r_a)^2.$$

Then $S(x_t) \ge 0$ and, whenever $\sum_{t<T} a_t > 0$,

$$f(x_{t+1}) \le f(x_t) - a_t S(x_t) + \frac{1}{2} C_0 a_t^2,$$

and

$$\min_{t<T} S(x_t) \le \frac{f(x_0) - f_{\inf} + \frac{1}{2} C_0 \sum_{t<T} a_t^2}{\sum_{t<T} a_t}.$$

Proof. Nonnegativity of S is the Frank–Wolfe gap inequality applied blockwise. For $h \in \mathcal{H}$, $v_h = -(r_h/\lambda_h) D_{N_h}(g_h)$ minimizes $\langle g_h, \cdot\rangle$ over C_h, so $G_h(x) = \langle g_h, x_h - v_h\rangle \ge 0$ whenever $x_h \in C_h$; feasibility holds for every t by $x_{h,0} \in C_h$, $a_t\bar\eta_h\lambda_h \le 1$, and the convexity argument of Theorem 9. For $a \in \mathcal{A}$, $N_a^*(g_a) \ge 0$ by definition of a dual norm. Hence $S(x_t) \ge 0$.
For $h \in \mathcal{H}$, $s_h = a_t\bar\eta_h\lambda_h(v_h - x_h)$ and feasibility gives $N_h(s_h) \le 2a_t\bar\eta_h r_h$. Its linear term is $-a_t\bar\eta_h\lambda_h G_h$. For $a \in \mathcal{A}$, $s_a = -a_t\bar\eta_a r_a D_{N_a}(g_a)$, with norm at most $a_t\bar\eta_a r_a$ and linear term $-a_t\bar\eta_a r_a N_a^*(g_a)$. Insert both into the joint smoothness bound and telescope. □

This proves a hybrid constrained/unconstrained stationarity statement for exact deterministic updates. It is not gradient descent on $f + \lambda\|x\|_2^2/2$ and does not by itself cover practical momentum, numerical approximations, or stochastic pretraining with decay. In summary, exact deterministic progress and inexact-oracle bounds hold under explicit smoothness and error assumptions, without implying unconditional convergence of practical stochastic pretraining.

I Extension to Encoder-Decoder Architectures (T5)

Before detailing the formal equivalence, it is helpful to establish the architectural context of models such as T5. Unlike the decoder-only architectures discussed previously, which employ a two-way weight tying scheme (sharing weights between the input token embedding and the pre-softmax output projection), T5 utilizes an encoder-decoder framework.

In this three-way tying setting, a single vocabulary table $E \in \mathbb{R}^{V \times d}$ is shared across three distinct operational roles:

- Encoder Lookup: Embedding the input sequence tokens into the encoder's continuous space.
- Decoder Lookup: Embedding the shifted target sequence tokens into the decoder.
- Output Classification: Projecting the final decoder hidden states back into the vocabulary space to generate softmax logits.

Intuitively, while this architecture introduces a third path for gradient flow, the encoder and decoder lookup operations are highly symmetric. From a geometric perspective, both act simply as identity row-selectors from E, constrained by the exact same bounds and radii. Because they serve identical mathematical roles in shaping the parameter space, the addition of the encoder lookup does not alter the fundamental topology of the optimization landscape. We can formalize this intuition by showing that the three-way alias norm collapses directly into the two-way case.

Proposition 8 (Equivalence of Three-Way and Two-Way Tying)**.** *Consider a fully shared T5-style vocabulary table $E \in \mathbb{R}^{V \times d}$ operating across the three roles described above. Because the encoder and decoder lookup roles share identical identity maps, norms, and radii, they constitute a geometrically inert duplicate role. Consequently, the three-way alias norm collapses exactly to the two-way tied intersection norm:*

$$N_{\text{alias}}^{\text{T5}}(U) = \max\{N_{\text{in}}(U), N_{\text{out},c}(U)\} = N_{\text{tie},c}(U)$$

Therefore, all geometric, oracle, dual, and symmetry results from the two-way decoder-only setting transfer verbatim. Furthermore, because shared redundancy is monotonic, adding a duplicate lookup role cannot reopen the head-shift redundancy quotient, ensuring $K_{\text{alias}}^{\text{T5}} = \{0\}$ remains true.

Proof. Index the three roles by $r \in \{\text{enc}, \text{dec}, \text{out}\}$ with role maps $A_{\text{enc}} = A_{\text{dec}} = A_{\text{out}}$, norms $N_{\text{enc}} = N_{\text{dec}} = N_{\text{in}}$ and $N_{\text{out}} = \|\cdot\|_\infty$, and radii $\rho_{\text{enc}} = \rho_{\text{dec}} = 1$, $\rho_{\text{out}} = c$. By the definition of the alias norm,

$$N_{\text{alias}}^{\text{T5}}(U) = \max\Big\{N_{\text{in}}(U),\, N_{\text{in}}(U),\, \tfrac{\|U\|_\infty}{c}\Big\} = \max\Big\{N_{\text{in}}(U),\, \tfrac{\|U\|_\infty}{c}\Big\} = N_{\text{tie},c}(U),$$

since a maximum is unchanged by repeating one of its arguments. The unit ball, the dual of Theorem 2, and the oracle of Theorem 4 depend on the roles only through this functional, so each coincides with its two-way counterpart. For the dual, the duplicated role contributes a third summand $\rho_{\text{enc}} N_{\text{in}}^*(Z_{\text{enc}})$ whose optimal value is attained at $Z_{\text{enc}} = 0$: replacing $(Z_{\text{enc}}, Z_{\text{dec}})$ by $(0, Z_{\text{enc}} + Z_{\text{dec}})$ preserves the constraint $\sum_r Z_r = B$ and cannot increase the objective, by the triangle inequality for N_{in}^* and the equality of the two radii. For the redundancy claim, Theorem 3 gives

$$K_{\text{alias}}^{\text{T5}} = K_{\text{enc}} \cap K_{\text{dec}} \cap K_{\text{out}} \subseteq K_{\text{dec}} \cap K_{\text{out}} = \{0\},$$

the last equality by Proposition 3. Adjoining a role can only shrink an intersection, which is the monotonicity asserted in the statement. □

I.1 Gradient Structure and Convergence

While the geometry remains unchanged, the gradient bookkeeping requires a minor structural extension to account for the additional lookup path:

- Combined Lookup Gradients: The total input-role gradient is the sum of the encoder and decoder lookup gradients, $G_{\text{in}}^{\text{T5}} = G_{\text{enc}} + G_{\text{dec}}$.
- Sparsity: This combined input gradient remains row-sparse and is supported strictly on the union of the encoder and decoder occupied token positions within the batch.

- Density: The output classification gradient G_{out} remains generically dense across all vocabulary rows.
- Convergence: Convergence guarantees transfer with identical proofs under standard block reindexing by simply treating the encoder and decoder hidden weights as an enlarged set of decoupled parameter blocks.

J Details of Experiments

J.1 NanoGPT

We include a NanoGPT-style decoder-only experiment to test AF-Muon in a standard tied input/output embedding language-model setting. The model has 12 layers, 6 attention heads, head dimension 128, hidden size 768, MLP hidden size 3072, and a 50,304-token vocabulary. The input and output vocabulary tables are tied: $E_{\text{in}} = W_{\text{out}}^{\top}$ Thus the shared table receives sparse input-lookup gradients and dense output-classifier gradients.

All runs use FP32 trainable parameters and optimizer state with BF16 autocast, constant learning rates, matrix learning rate 0.02, auxiliary/vector learning rate 3×10^{-4}, matrix weight decay 0.1, gradient clipping at norm 1.0, batch size 512, context length 1024, and 2.62B training tokens. The Hybrid Muon baseline applies auxiliary AdamW weight decay 0.01 to the tied and one-dimensional auxiliary parameters. AF-Muon and SCION-style Sign use zero auxiliary/tied weight decay.

Table 2: Per-seed NanoGPT-style decoder-only results. All runs use tied input/output embeddings and 2.67B training tokens. The final row reports a paired two-sided t-test over validation loss between AF-Muon and the strongest non-AF baseline by mean validation loss. The validation-loss entry is the mean paired gain, computed as baseline minus AF-Muon, so positive values indicate that AF-Muon is better.

Method	Seed	Train loss ↓	Validation loss ↓	Validation PPL ↓
Hybrid Muon	43	3.5182	3.5212	33.83
Hybrid Muon	44	3.5133	3.5152	33.62
Hybrid Muon	45	3.5134	3.5165	33.67
SCION-style Sign	43	3.5185	3.5272	34.03
SCION-style Sign	44	3.5001	3.5046	33.27
SCION-style Sign	45	3.5187	3.5247	33.94
AF-Muon	43	3.4910	3.4936	32.91
AF-Muon	44	3.4896	3.4898	32.78
AF-Muon	45	3.4889	3.4921	32.86
AF-Muon vs. Hybrid Muon (paired t-test)		validation-loss gain $= 0.0258$, $p = 0.0013$		

Table 3: Final NanoGPT tied-vocabulary diagnostics. Hybrid Muon uses auxiliary AdamW weight decay 0.01; SCION-style Sign and AF-Muon use zero auxiliary/tied weight decay.

Method	Seed	Tied RMS	Tied max	Matrix RMS	Aux WD
Hybrid Muon	43	0.0427	0.2500	0.0463	0.01
Hybrid Muon	44	0.0429	0.2543	0.0465	0.01
Hybrid Muon	45	0.0431	0.2568	0.0464	0.01
SCION-style Sign	43	0.5528	7.8773	0.0473	0.00
SCION-style Sign	44	0.6238	7.8289	0.0466	0.00
SCION-style Sign	45	0.5548	7.8897	0.0471	0.00
AF-Muon	43	0.3821	5.8463	0.0469	0.00
AF-Muon	44	0.3895	9.2696	0.0469	0.00
AF-Muon	45	0.3849	9.4472	0.0468	0.00

The diagnostics show that AF-Muon improves validation loss while keeping the shared tied table substantially smaller than the SCION-style Sign endpoint in RMS norm. Hybrid Muon keeps the tied

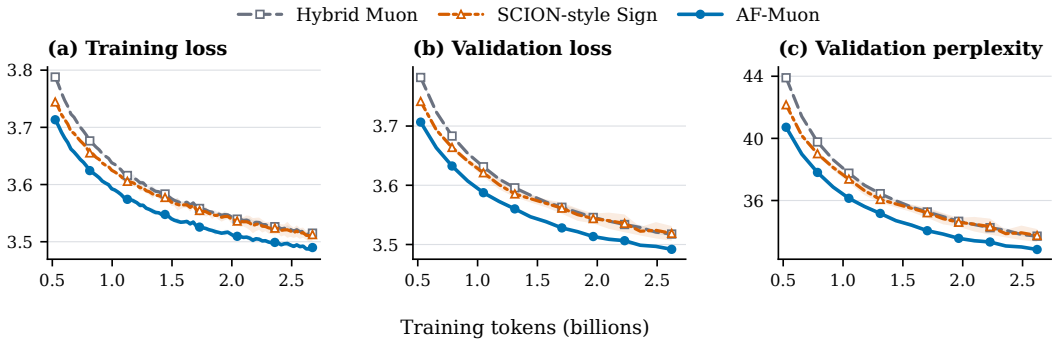


Figure 6: NanoGPT-style decoder-only training dynamics on FineWeb. Curves show the mean over three paired seeds, and shaded regions denote ± 1 standard error of the mean.

table much smaller because its auxiliary AdamW branch includes weight decay, but this additional radial shrinkage does not close the performance gap. This supports the interpretation that the finite-cap tied-table geometry is not merely acting as uncontrolled norm growth: it gives a better tied-table update direction under the matched matrix-update protocol.

J.2 SmolLM2-135M

We include a SmolLM2-135M decoder-only experiment to test AF-Muon in a pretrained-tokenizer tied input/output embedding language-model setting. The model has approximately 134.5M trainable parameters and uses the SmolLM2 tokenizer and tied input/output vocabulary table: $E_{\text{in}} = W_{\text{out}}^{\top}$ Thus the shared table receives sparse input-lookup gradients and dense output-classifier gradients.

Table 4: Per-seed SmolLM2-135M decoder-only results. All runs use tied input/output embeddings and 2.49B training tokens. The final row reports a paired two-sided t-test over validation loss between AF-Muon and the strongest non-AF baseline by mean validation loss; the gain is baseline minus AF-Muon, so positive values indicate that AF-Muon is better.

Method	Seed	Train loss ↓	Validation loss ↓	Validation PPL ↓
Hybrid Muon	43	3.5118	3.5047	33.27
Hybrid Muon	44	3.5082	3.5058	33.31
Hybrid Muon	45	3.5063	3.5088	33.41
SCION-style Sign	43	3.4777	3.4744	32.28
SCION-style Sign	44	3.4757	3.4663	32.02
SCION-style Sign	45	3.4649	3.4712	32.17
AF-Muon	43	3.4451	3.4405	31.20
AF-Muon	44	3.4460	3.4461	31.38
AF-Muon	45	3.4449	3.4473	31.41
AF-Muon vs. SCION-style (paired t-test)		validation-loss gain $= 0.0260$, $p = 0.0239$		

All runs use FP32 trainable parameters and optimizer state with BF16 autocast, constant learning rates, matrix learning rate 0.02, auxiliary/vector learning rate 3×10^{-4}, matrix weight decay 0.1, gradient clipping at norm 1.0, batch size 512, context length 1024, and 2.49B training tokens. The matched large-batch protocol uses micro-batch size 32 and gradient accumulation 16, giving 524k tokens per optimizer update. The Hybrid Muon baseline applies auxiliary/tied AdamW weight decay 0.01 to the tied and one-dimensional auxiliary parameters. AF-Muon and SCION-style Sign use zero auxiliary/tied weight decay.

The diagnostics show that AF-Muon improves validation loss while keeping the shared tied table smaller in RMS norm than the SCION-style Sign endpoint. Hybrid Muon keeps the tied table much smaller because its auxiliary AdamW branch includes weight decay, but this additional radial shrinkage does not close the performance gap. This supports the interpretation that the finite-cap

Table 5: Final SmolLM2-135M tied-vocabulary diagnostics. Hybrid Muon uses auxiliary/tied AdamW weight decay 0.01; SCION-style Sign and AF-Muon use zero auxiliary/tied weight decay.

Method	Seed	Tied RMS	Tied max	Matrix RMS	Aux WD
Hybrid Muon	43	0.0549	0.3561	0.0561	0.01
Hybrid Muon	44	0.0552	0.3029	0.0559	0.01
Hybrid Muon	45	0.0551	0.3498	0.0560	0.01
SCION-style Sign	43	0.6611	10.0026	0.0562	0.00
SCION-style Sign	44	0.6524	10.0348	0.0560	0.00
SCION-style Sign	45	0.6668	10.0104	0.0560	0.00
AF-Muon	43	0.4404	9.3857	0.0566	0.00
AF-Muon	44	0.4457	12.9820	0.0567	0.00
AF-Muon	45	0.4415	13.6058	0.0566	0.00

tied-table geometry is not merely acting as uncontrolled norm growth: it gives a better tied-table update direction under the matched matrix-update protocol.

J.3 QWEN2.5-0.5B

We include Qwen2.5-0.5B to test AF-Muon on a larger decoder-only architecture with tied input/output embeddings. The model uses the Qwen2.5-0.5B tokenizer and a shared input/output vocabulary table, $E_{\text{in}} = W_{\text{out}}^{\top}$ As in the NanoGPT and SmolLM2 experiments, this shared table receives both sparse input-lookup gradients and dense output-classifier gradients.

All runs use FP32 trainable parameters and optimizer state with BF16 autocast, constant learning rates, matrix learning rate 0.02, auxiliary/vector learning rate 3×10^{-4}, matrix weight decay 0.1, gradient clipping at norm 1.0, context length 1024, and 524,288 tokens per optimizer update. The short three-seed runs use 1B training tokens. Hybrid Muon applies auxiliary AdamW weight decay 0.01 to the tied and one-dimensional auxiliary parameters, while SCION-style Sign and AF-Muon use zero auxiliary/tied weight decay.

Table 6: Per-seed Qwen2.5-0.5B short-horizon results. Values are reported at the last evaluation before the 1B-token budget, corresponding to 0.918B training tokens. The final row reports a paired two-sided t-test over validation loss between AF-Muon and the strongest non-AF baseline by mean validation loss; the gain is baseline minus AF-Muon, so positive values indicate that AF-Muon is better.

Method	Seed	Train loss ↓	Validation loss ↓	Validation PPL ↓
Hybrid Muon	43	3.6186	3.6349	37.90
Hybrid Muon	44	3.6293	3.6406	38.12
Hybrid Muon	45	3.6329	3.6424	38.18
SCION-style Sign	43	3.5935	3.6075	36.87
SCION-style Sign	44	3.5990	3.6133	37.09
SCION-style Sign	45	3.6026	3.6119	37.04
AF-Muon	43	3.5660	3.5805	35.89
AF-Muon	44	3.5715	3.5831	35.98
AF-Muon	45	3.5763	3.5858	36.08
AF-Muon vs. SCION-style (paired t-test)		validation-loss gain $= 0.0278$, $p = 0.0020$		

At the near-1B evaluation point, AF-Muon achieves the lowest validation loss on all three paired Qwen2.5-0.5B seeds. The paired improvements are 0.0562 ± 0.0016 validation-loss points over Hybrid Muon and 0.0278 ± 0.0022 points over SCION-style Sign. This ordering is also preserved in the seed-43 long-horizon continuation in Appendix J.3.1.

The short-horizon checkpoint diagnostics show the same qualitative pattern as the long-horizon Qwen continuation: Hybrid Muon keeps the tied table smallest because of auxiliary/tied AdamW

Table 7: Final Qwen2.5-0.5B tied-table diagnostics for the 1B short-horizon seed-44 and seed-45 runs. Hybrid Muon uses auxiliary AdamW weight decay 0.01; SCION-style Sign and AF-Muon use zero auxiliary/tied weight decay. Seed-43 checkpoint diagnostics are reported separately for the long-horizon continuation in Table 9.

Method	Seed	Tied RMS	Tied max	Matrix RMS	Aux WD
Hybrid Muon	44	0.0481	0.3676	0.0435	0.01
Hybrid Muon	45	0.0488	0.3256	0.0433	0.01
SCION-style Sign	44	0.2732	2.6204	0.0440	0.00
SCION-style Sign	45	0.2666	2.6204	0.0439	0.00
AF-Muon	44	0.1997	2.6786	0.0445	0.00
AF-Muon	45	0.1934	3.0957	0.0444	0.00

weight decay, SCION-style Sign produces the largest tied-table RMS norm, and AF-Muon remains intermediate while achieving the best validation loss.

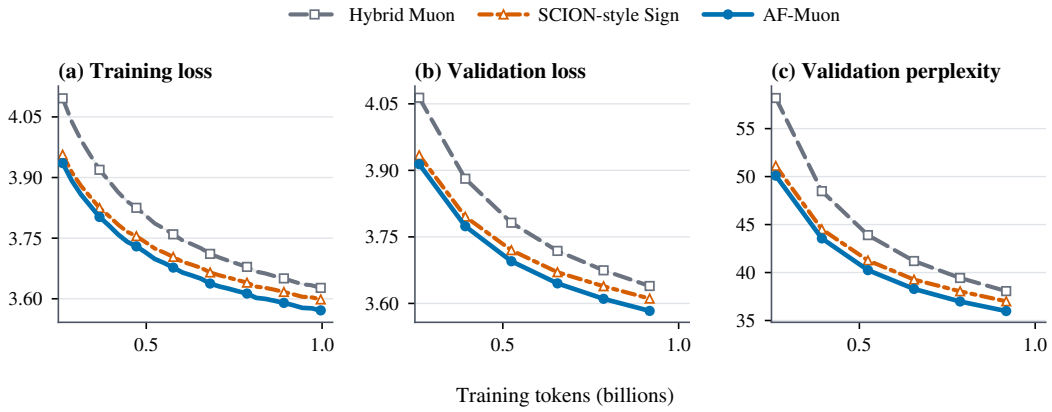


Figure 7: Qwen2.5-0.5B training dynamics on FineWeb. Curves show the mean over three paired seeds, and shaded regions denote ± 1 standard error of the mean. The plot shows the short-horizon region used for the main three-seed comparison, omitting the earliest training region for readability. All methods use tied input/output embeddings, FP32 trainable parameters and optimizer state with BF16 autocast, matrix weight decay 0.1, and the same matched large-batch constant-learning-rate protocol.

J.3.1 Qwen2.5-0.5B Long-horizon Continuation

In the seed-43 continuation trained to 6.40B tokens, AF-Muon obtains the lowest validation loss at the last scheduled evaluation, 6.29B tokens, followed by SCION-style Sign and then Hybrid Muon. AF-Muon improves over Hybrid Muon by 0.0316 validation-loss points and over SCION-style Sign by 0.0138 points at that evaluation point. The final-checkpoint diagnostics show the same qualitative tied-table norm ordering observed in the other tied-vocabulary experiments: Hybrid Muon keeps the tied table smallest because its auxiliary AdamW branch applies weight decay, SCION-style Sign produces the largest tied-table norm, and AF-Muon remains intermediate while obtaining the best endpoint loss. As with the Llama-3.2-1B continuation, this is a single-seed stability check for the short-horizon three-seed comparison rather than an independent multi-seed estimate.

Table 8: Qwen2.5-0.5B long-horizon seed-43 endpoint results. The run continues the same seed used in the paired short-horizon comparison to 6.40B training tokens; the validation endpoint below is the last scheduled evaluation, at 6.29B tokens. This single-seed continuation is used as a stability check, not as a three-seed estimate.

Optimizer	Train loss ↓	Validation loss ↓	Validation PPL ↓
Hybrid Muon	3.4254	3.4274	30.80
SCION-style Sign	3.4068	3.4096	30.25
AF-Muon	**3.3922**	**3.3957**	**29.84**

Table 9: Final Qwen2.5-0.5B long-horizon seed-43 checkpoint diagnostics. The tied-table and matrix norms are measured from the final model-only checkpoint at 6.40B training tokens. Hybrid Muon uses auxiliary/tied AdamW weight decay 0.01; SCION-style Sign and AF-Muon use zero auxiliary/tied weight decay. All runs use FP32 trainable parameters and optimizer state with BF16 autocast.

Optimizer	Tied RMS	Tied max	Matrix RMS	Matrix WD	Aux./tied WD	Precision
Hybrid Muon	0.0553	0.4511	0.0439	0.1	0.01	FP32 + BF16
SCION-style Sign	0.5815	7.4900	0.0450	0.1	0.00	FP32 + BF16
AF-Muon	0.3868	6.5167	0.0452	0.1	0.00	FP32 + BF16

J.4 Llama-3.2-1B Decoder-only Results

We include Llama-3.2-1B to test AF-Muon on a larger decoder-only architecture with a high-capacity tied vocabulary table. The model uses the Llama-3.2-1B tokenizer and tied input/output embeddings, $E_{\text{in}} = W_{\text{out}}^{\top}$ As in the other decoder-only language-model experiments, the shared table receives both sparse input-lookup gradients and dense output-classifier gradients.

All runs use FP32 trainable parameters and optimizer state with BF16 autocast, constant learning rates, matrix learning rate 0.02, auxiliary/vector learning rate 3×10^{-4}, matrix weight decay 0.1, gradient clipping at norm 1.0, context length 1024, and 524,288 tokens per optimizer update. The short three-seed runs use 2B training tokens. Hybrid Muon applies auxiliary/tied AdamW weight decay 0.01 to the tied and one-dimensional auxiliary parameters, while SCION-style Sign and AF-Muon use zero auxiliary/tied weight decay. The long-horizon seed-43 continuation uses the same settings and extends training to 9.98B tokens.

Table 10: Per-seed Llama-3.2-1B short-horizon results. Values are reported at the final available evaluation near the 2B-token budget. Seed 43 is read from the long-horizon continuation at the last pre-2B evaluation point, 1.966B tokens; seeds 44 and 45 are read from the 2B short-horizon runs. The final row reports a paired two-sided t-test over validation loss between AF-Muon and the strongest non-AF baseline by mean validation loss; the gain is baseline minus AF-Muon, so positive values indicate that AF-Muon is better.

Method	Seed	Train loss ↓	Validation loss ↓	Validation PPL ↓
Hybrid Muon	43	3.4927	3.5348	34.29
Hybrid Muon	44	3.4908	3.5400	34.47
Hybrid Muon	45	3.4962	3.5415	34.52
SCION-style Sign	43	3.4755	3.5139	33.58
SCION-style Sign	44	3.4750	3.5280	34.06
SCION-style Sign	45	3.4809	3.5256	33.97
AF-Muon	43	3.4600	3.5012	33.15
AF-Muon	44	3.4565	3.5107	33.47
AF-Muon	45	3.4617	3.5051	33.28
AF-Muon vs. SCION-style (paired t-test)		validation-loss gain $= 0.0168$, $p = 0.0176$		

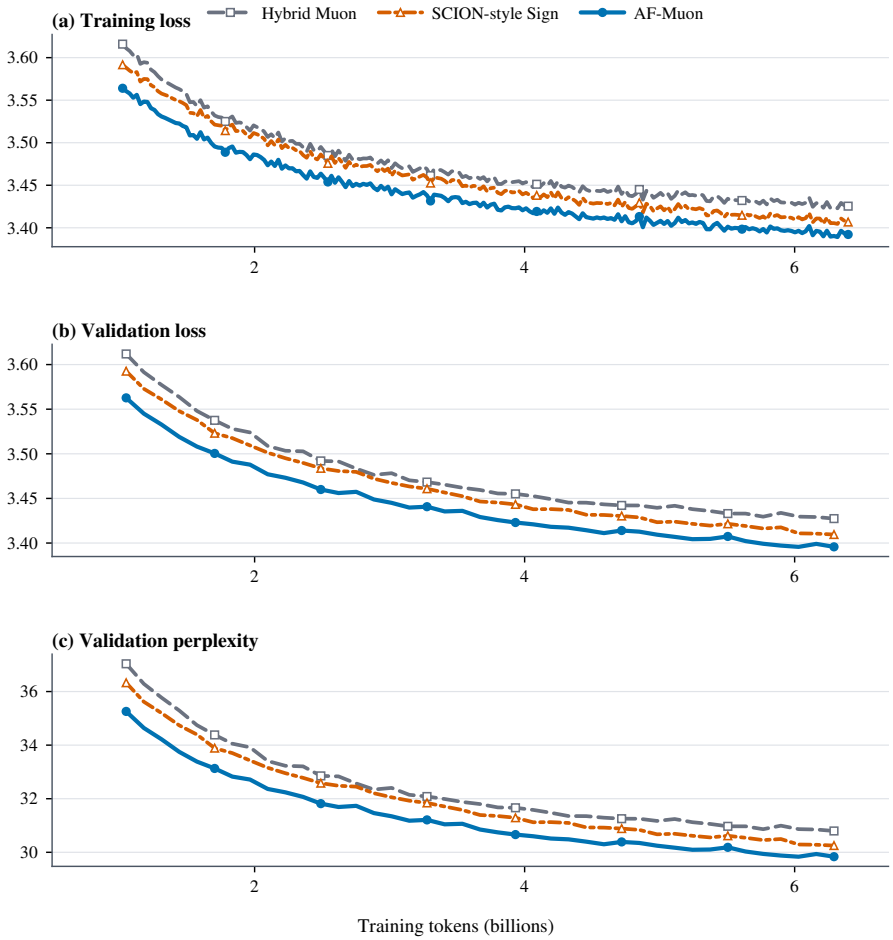


Figure 8: Qwen2.5-0.5B long-horizon seed-43 continuation. The three optimizer arms are trained under the same matched large-batch protocol as the short-horizon runs. The plotted curves omit the earliest training region and begin after 1B tokens to focus on the long-horizon behavior. This figure is a single-seed stability check, not a replacement for the three-seed short-horizon comparison. AF-Muon preserves the best validation loss at the last scheduled evaluation point, 6.29B tokens.

Table 11: Final Llama-3.2-1B tied-table diagnostics for the 2B short-horizon seed-44 and seed-45 runs. Hybrid Muon uses auxiliary/tied AdamW weight decay 0.01; SCION-style Sign and AF-Muon use zero auxiliary/tied weight decay. Seed-43 checkpoint diagnostics are reported separately for the long-horizon continuation in Table 13.

Method	Seed	Tied RMS	Tied max	Matrix RMS	Aux WD
Hybrid Muon	44	0.0311	0.3537	0.0287	0.01
Hybrid Muon	45	0.0310	0.2848	0.0287	0.01
SCION-style Sign	44	0.1690	2.2859	0.0287	0.00
SCION-style Sign	45	0.1718	2.3012	0.0287	0.00
AF-Muon	44	0.1129	2.2891	0.0290	0.00
AF-Muon	45	0.1125	1.3577	0.0290	0.00

The checkpoint diagnostics show the same tied-table norm ordering as in the Qwen2.5-0.5B runs: Hybrid Muon keeps the tied table smallest because its auxiliary AdamW branch applies weight decay,

SCION-style Sign produces the largest tied-table RMS norm, and AF-Muon remains intermediate while achieving the best validation loss.

J.4.1 LLAMA-3.2-1B LONG-HORIZON CONTINUATION

To test whether the short-horizon optimizer ordering persists beyond the three-seed 2B-token budget, we continue seed 43 to 9.98B training tokens. This run is not used as an independent three-seed estimate; instead, it checks whether the behavior observed in the paired short-horizon runs remains stable when one representative seed is trained for a substantially larger token budget.

Table 12: Llama-3.2-1B long-horizon seed-43 endpoint results.

Optimizer	Train loss ↓	Validation loss ↓	Validation PPL ↓
Hybrid Muon	3.3848	3.4251	30.73
SCION-style Sign	3.3534	3.3951	29.82
AF-Muon	**3.3504**	**3.3912**	**29.70**

Table 13: Final Llama-3.2-1B long-horizon seed-43 diagnostics. All runs use FP32 trainable parameters and optimizer state with BF16 autocast.

Optimizer	Grad norm	Tied RMS	Tied max	Matrix WD	Aux. WD	Tied WD	Precision
Hybrid Muon	0.3709	0.0368	0.3828	0.1	0.01	0.01	FP32 + BF16
SCION-style Sign	0.3029	0.3656	6.1341	0.1	0.00	0.00	FP32 + BF16
AF-Muon	0.3084	0.2463	2.8759	0.1	0.00	0.00	FP32 + BF16

In the 9.98B-token seed-43 continuation shown in Figure 9, AF-Muon obtains the lowest final validation loss, with SCION-style Sign close behind and both AdamW-free tied-table arms clearly ahead of Hybrid Muon. The final AF-Muon advantage over SCION-style Sign is small, but the ordering is preserved at the end of the long-horizon run. We therefore use this continuation as a stability check for the three-seed short-horizon comparison, rather than as a separate multi-seed estimate.

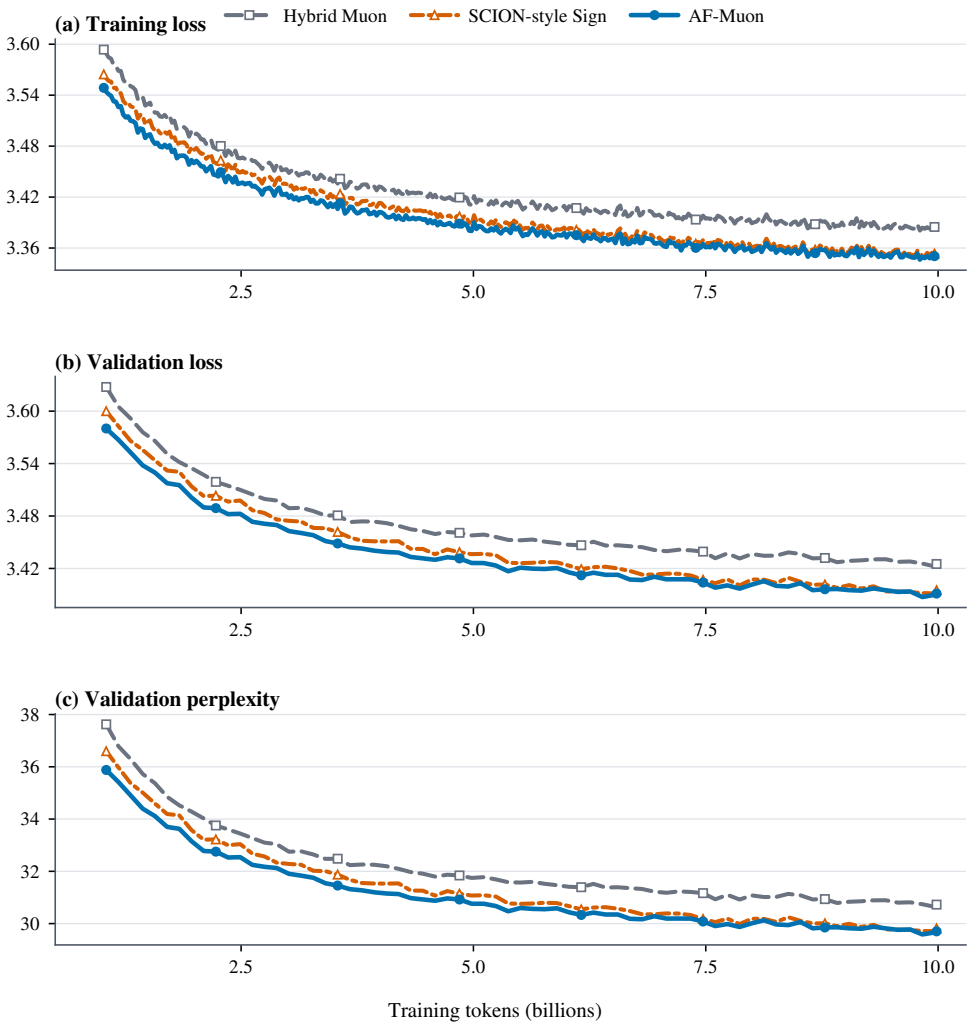


Figure 9: Llama-3.2-1B long-horizon seed-43 continuation. The curves omit the earliest training region and begin after 1B tokens to focus on long-horizon behavior. This is a single-seed stability check, not a replacement for the three-seed short-horizon comparison. AF-Muon preserves the best endpoint validation loss at 9.98B tokens, while both AdamW-free tied-table arms remain ahead of Hybrid Muon.

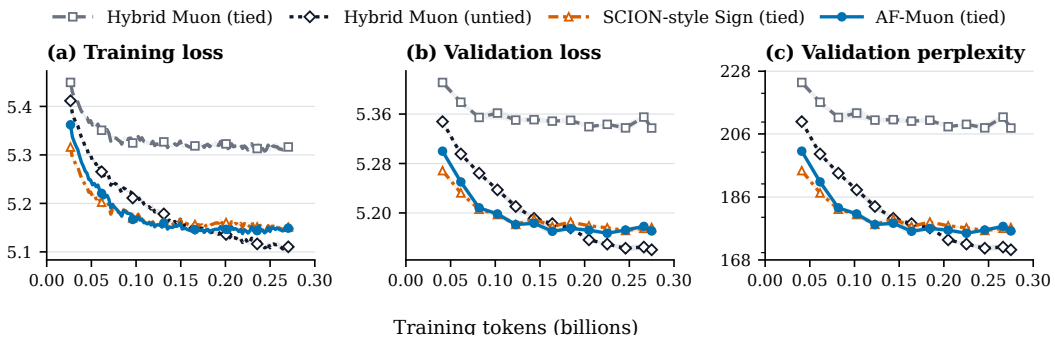


Figure 10: Small T5-style training dynamics under shared-vocabulary and untied-vocabulary topologies. Curves show the mean over three seeds. The untied Hybrid Muon control has nearly twice as many parameters, because the encoder embedding, decoder embedding, and output head are separate.

J.5 Shared-vocabulary encoder–decoder experiments

We include two encoder–decoder experiments to test AF-Muon in a shared-parameter topology different from decoder-only language modeling. In both cases, one vocabulary table is used for encoder input, decoder input, and output classification, so the table receives sparse encoder-lookup gradients, sparse decoder-lookup gradients, and dense output-head gradients. They answer complementary questions:

- **Small T5-style:** Does AF-Muon work in a controlled encoder–decoder setup, including a tied-versus-untied comparison?
- **T5Gemma2-style:** Does the optimizer recipe transfer to a different, larger encoder–decoder configuration?

Small T5-style diagnostic. The small controlled model is a T5-style encoder–decoder trained with a prefix-to-continuation objective on FineWeb Table 15. The encoder input length and decoder target length are both 256 tokens. The model has 4 encoder layers, 4 decoder layers, hidden size 512, feed-forward size 2048, 8 attention heads, and a 49,152-token vocabulary. Unless otherwise stated, runs use FP32 trainable parameters and optimizer state with BF16 autocast, constant learning rates, matrix weight decay 0.1, and no learning-rate schedule. In the fully shared setting, $E_{\text{enc}} = E_{\text{dec}} = W_{\text{out}}^{\top}$.

Table 14: Single-seed ρ sweep for AF-Muon on small T5-style shared-vocabulary encoder–decoder with a fully shared vocabulary table. All runs use 110M target tokens, matrix weight decay 0.1, and zero auxiliary/tied weight decay. We select $\rho_{\text{out}}/\rho_{\text{hid}} = 1$ for the main three-seed experiment.

Method	$\rho_{\text{out}}/\rho_{\text{hid}}$	Tied LR	Eval loss ↓	Eval PPL ↓	Shared RMS	Shared max
Hybrid Muon	–	–	5.3468	209.93	0.142	3.19
AF-Muon	1	1.9531×10^{-5}	**5.2652**	**193.48**	0.318	7.51
AF-Muon	5	9.7656×10^{-5}	5.3536	211.36	0.721	8.99
AF-Muon	10	1.9531×10^{-4}	5.4220	226.34	1.136	20.43
AF-Muon	60	1.1719×10^{-3}	5.4864	241.38	2.157	37.02

The sweep Table 14 shows that the decoder-only radius ratio used in the causal language model experiments is too large for this encoder–decoder topology: as $\rho_{\text{out}}/\rho_{\text{hid}}$ increases, the shared table grows rapidly and validation loss degrades. We therefore use $\rho_{\text{out}}/\rho_{\text{hid}} = 1$ for the main small T5-style comparison.

The fully untied Hybrid Muon control is not parameter matched: it uses separate encoder, decoder, and output vocabulary tables, increasing the model from 54.6M to 104.9M parameters. This control isolates the cost of enforcing a single shared table. Untying improves Hybrid Muon, but within the stricter fully shared setting AF-Muon is the best tied method by mean validation loss, with only a small and statistically insignificant margin over SCION-style Sign.

Table 15: Per-seed small T5-style shared-vocabulary encoder–decoder results for the selected shared-vocabulary setting. All runs use 275M target tokens. Hybrid Muon uses auxiliary AdamW weight decay 0.01; SCION-style Sign and AF-Muon use zero auxiliary/tied weight decay. The table reports paired two-sided t-tests over validation loss comparing AF-Muon with Hybrid Muon and with the strongest non-AF baseline – in the fully shared setting – by mean validation loss.

Method	Topology	Seed	Validation loss ↓	Validation PPL ↓
Hybrid Muon		43	5.3372	207.94
Hybrid Muon		44	5.3272	205.86
Hybrid Muon		45	5.3476	210.11
SCION-style Sign		43	5.1643	174.91
SCION-style Sign	Fully shared	44	5.1714	176.17
SCION-style Sign		45	5.1930	180.01
AF-Muon		43	5.1700	175.91
AF-Muon		44	5.1745	176.72
AF-Muon		45	5.1678	175.53
AF-Muon vs. Hybrid Muon (paired t-test)			validation-loss gain $= 0.1666$, $p = 0.0022$	
AF-Muon vs. SCION-style Sign (paired t-test)			validation-loss gain $= 0.0055$, $p = 0.6361$	
Hybrid Muon		43	5.1355	169.96
Hybrid Muon	Fully untied	44	5.1419	171.05
Hybrid Muon		45	5.1437	171.35

T5Gemma2-style transfer experiment. We also run a larger T5Gemma2-style random-init configuration based on Zhang et al. (2025a). This experiment tests whether the same optimizer recipe transfers beyond the small controlled T5-style diagnostic. The run uses the Hugging Face `google/t5gemma-2-270m-270m` configuration from random initialization, a shared 262,208-row vocabulary table, source length 256, target length 256, and 750M target tokens. The effective target-token batch is 524,288 tokens per update, using microbatch size 16 and 128 gradient-accumulation steps. All trainable parameters are FP32 with BF16 autocast. We use the main $\rho_{\text{out}}/\rho_{\text{hid}} = 60$ setting, matrix weight decay 0.1, Hybrid Muon auxiliary weight decay 0.01, and zero auxiliary/tied weight decay for SCION-style Sign and AF-Muon.

Table 16: T5Gemma2-style shared-vocabulary encoder–decoder results at 750M target tokens. Runs use the random-init `google/t5gemma-2-270m-270m` configuration with a shared 262k-row vocabulary table. Per-seed rows report validation metrics; summary rows report mean ± SD over paired seeds. Paired tests compare validation loss against AF-Muon; positive gain means AF-Muon is better.

Method	Seed/Summary	Train loss ↓	Validation loss ↓	Validation PPL ↓
Hybrid Muon	43	3.6366	3.6501	38.48
Hybrid Muon	44	3.6455	3.6578	38.77
Hybrid Muon	45	3.6407	3.6566	38.73
Hybrid Muon	Mean ± SD	3.6409 ± 0.0044	3.6548 ± 0.0041	38.66 ± 0.16
SCION-style Sign	43	3.5705	3.5871	36.13
SCION-style Sign	44	3.5703	3.5867	36.11
SCION-style Sign	45	3.5685	3.5840	36.02
SCION-style Sign	Mean ± SD	3.5698 ± 0.0011	3.5859 ± 0.0017	36.09 ± 0.06
AF-Muon	43	3.5561	3.5723	35.60
AF-Muon	44	3.5579	3.5742	35.67
AF-Muon	45	3.5578	3.5738	35.65
AF-Muon	Mean ± SD	**3.5573 ± 0.0010**	**3.5735 ± 0.0010**	**35.64 ± 0.04**
AF-Muon vs. Hybrid Muon		validation-loss gain $= 0.0813$, paired $p = 4.87 \times 10^{-4}$		
AF-Muon vs. SCION-style Sign		validation-loss gain $= 0.0125$, paired $p = 0.0112$		

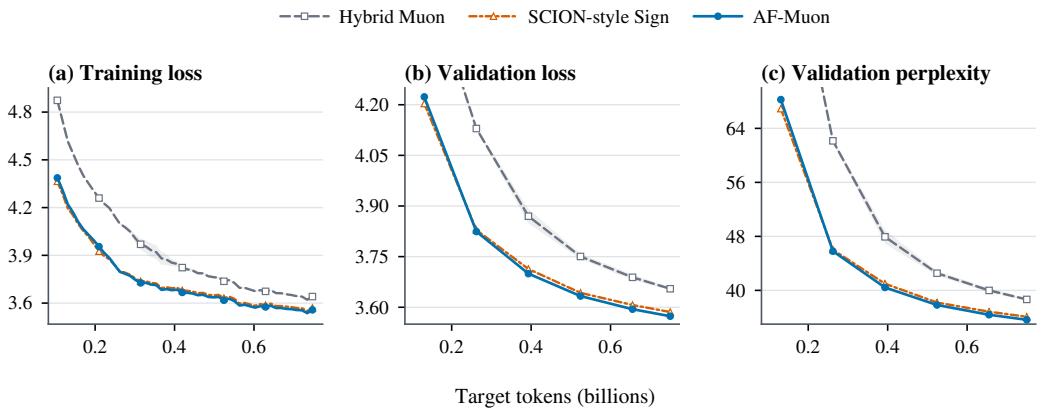


Figure 11: T5Gemma2-style shared-vocabulary encoder–decoder training dynamics after 100M target tokens. Curves show the mean over three paired seeds for Hybrid Muon, SCION-style Sign, and AF-Muon.

In the completed evaluated seeds, AF-Muon obtains the lowest validation loss and perplexity. The AF-Muon–Sign gap is larger than in the small T5-style diagnostic, but the two experiments differ in architecture, vocabulary configuration, and training budget, so this comparison does not isolate vocabulary size or model size as the cause. The result is therefore best interpreted as evidence that the AF-Muon tied-table recipe transfers to a larger encoder–decoder configuration.

J.6 NanoGPT-MoE

We include a tied-embedding mixture-of-experts ablation to test whether the AF-Muon tied-table update remains effective when the decoder-only model uses sparse conditional computation inside the transformer blocks. Starting from the NanoGPT-style tied-vocabulary setting, we replace each dense MLP with a top-1 router over four expert MLPs. Each expert uses the same hidden expansion as the dense NanoGPT MLP. The input and output vocabulary tables remain tied, so the shared table receives sparse input-lookup gradients and dense output-classifier gradients:

$$E_{\text{in}} = W_{\text{out}}^{\top}.$$

All runs use the same FineWeb token cache and 50,304-token tokenizer as the NanoGPT tied-vocabulary experiment. The model has 12 layers, 6 attention heads, head dimension 128, hidden size 768, four top-1 routed experts per layer, and MLP hidden size 3072 per expert. We train for 2.62B tokens across three paired seeds with FP32 trainable parameters and optimizer state, BF16 autocast, batch size 512, context length 1024, matrix learning rate 0.02, auxiliary/vector learning rate 3×10^{-4}, matrix weight decay 0.1, and gradient clipping at norm 1.0. The router auxiliary load-balancing coefficient is 0.01 for all optimizer arms, and the router z-loss coefficient is zero. The optimizer partition is matched across arms. Matrix parameters, including attention, expert, and router matrices, use Muon in all three arms. The only difference is the update applied to the tied table and remaining auxiliary parameters: Hybrid Muon uses its auxiliary AdamW branch with auxiliary/tied weight decay 0.01, SCION-style Sign uses the Sign endpoint with zero auxiliary/tied weight decay, and AF-Muon uses the finite-cap tied-table update with zero auxiliary/tied weight decay.

In the tied-embedding MoE setting, AF-Muon consistently achieves the lowest validation loss. SCION-style Sign is competitive early in training, but Hybrid Muon overtakes it later; at the final checkpoint AF-Muon improves over both Hybrid Muon and SCION-style Sign. Quantitatively, AF-Muon improves validation loss over Hybrid Muon by 0.0224 ± 0.0023 paired loss points and over SCION-style Sign by 0.0452 ± 0.0019. The router diagnostics show that this gain is not caused by expert collapse: all methods keep all four experts active, with router entropy near one and low expert-load coefficient of variation.

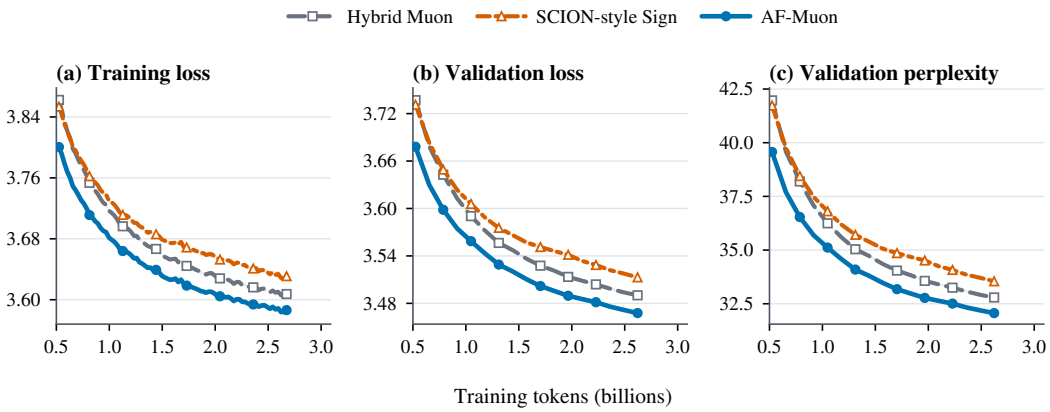


Figure 12: Tied-embedding NanoGPT-MoE training dynamics on FineWeb. Curves show the mean over three paired seeds, and shaded regions denote ± 1 standard error of the mean. All methods use four top-1 routed experts per layer and the same router auxiliary load-balancing coefficient.

Table 17: Tied-embedding NanoGPT-MoE results on FineWeb for 2.62B tokens across three paired seeds. We report mean $\pm$ SD over seeds. The model uses four top-1 routed experts per layer and a router auxiliary load-balancing coefficient of 0.01.

Optimizer	Train loss ↓	Validation loss ↓	Validation PPL ↓
Hybrid Muon	3.4873 ± 0.0031	3.4902 ± 0.0019	32.79 ± 0.06
SCION-style Sign	3.5106 ± 0.0033	3.5129 ± 0.0025	33.55 ± 0.08
AF-Muon	$\mathbf{3.4662 \pm 0.0011}$	$\mathbf{3.4677 \pm 0.0007}$	$\mathbf{32.06 \pm 0.02}$

J.7 Tied ImageGPT-style ImageNet-32

We include a cross-domain tied-table ablation on autoregressive image modeling to test whether the AF-Muon tied-embedding behavior also appears outside language. The model follows an ImageGPT-style causal-transformer setup on ImageNet-32, but with an intentionally tied input/output token table. This is therefore a controlled tied-table optimizer probe, not a faithful reproduction of the original untied ImageGPT setup.

Images are represented as sequences of 1024 discrete color tokens. We use a deterministic RGB554 tokenizer, which maps each 32-bit RGB pixel to one of $2^{14} = 16{,}384$ color tokens. We add one start-of-sequence token, giving a total vocabulary size of 16,385. The input sequence is formed by prepending the start token and shifting the image tokens right, while the output predicts only the 16,384 color tokens. Thus the color-token rows of the shared table receive both sparse input-lookup gradients and dense output-classifier gradients, while the start-token row receives only sparse input gradients.

The model has 8 transformer layers, 8 attention heads, head dimension 64, hidden size 512, and MLP hidden size 2048. All runs use FP32 trainable parameters and optimizer state with BF16 autocast, batch size 512, micro-batch size 256, sequence length 1024, matrix learning rate 0.02, auxiliary/vector learning rate 3×10^{-4}, matrix weight decay 0.1, momentum 0.95, and 1.31B image tokens. Hybrid Muon uses auxiliary/tied AdamW weight decay 0.01. SCION-style Sign and AF-Muon use zero auxiliary/tied weight decay. Because the RGB554 tokenizer is lossy and task-specific, the absolute bits/dim values are used only for within-experiment optimizer comparisons.

AF-Muon achieves the best validation loss and bits/dim across the three paired seeds. Relative to Hybrid Muon, AF-Muon improves validation loss by 0.0103 ± 0.0014 paired loss points and validation bits/dim by 0.0049 ± 0.0007. Relative to SCION-style Sign, AF-Muon improves validation loss by 0.0469 ± 0.0064 and validation bits/dim by 0.0226 ± 0.0031. The Sign endpoint is substantially

Table 18: Per-seed tied-embedding NanoGPT-MoE results. All runs use 2.62B training tokens, four top-1 routed experts per layer, and the same router auxiliary loss. The final row reports a paired two-sided t-test over validation loss between AF-Muon and the strongest non-AF baseline by mean validation loss; the gain is baseline minus AF-Muon, so positive values indicate that AF-Muon is better.

Method	Seed	Validation loss ↓	Validation PPL ↓
Hybrid Muon	43	3.4883	32.73
Hybrid Muon	44	3.4901	32.79
Hybrid Muon	45	3.4921	32.85
SCION-style Sign	43	3.5112	33.49
SCION-style Sign	44	3.5158	33.64
SCION-style Sign	45	3.5117	33.50
AF-Muon	43	3.4676	32.06
AF-Muon	44	3.4685	32.09
AF-Muon	45	3.4671	32.04
AF-Muon vs. Hybrid Muon (paired t-test)		validation-loss gain $= 0.0224$, $p = 0.0034$	

Table 19: Final tied-table and router diagnostics for the tied-embedding NanoGPT-MoE ablation. Values are mean ± SD over three paired seeds.

Method	Tied RMS	Tied max	Router entropy ↑	Max expert frac. ↓	Router CV ↓	Unused experts
Hybrid Muon	0.0424 ± 0.0001	0.2656 ± 0.0120	0.9996 ± 0.0001	0.2600 ± 0.0020	0.0290 ± 0.0056	0.00 ± 0.00
SCION-style Sign	0.6357 ± 0.0161	7.8549 ± 0.0381	0.9978 ± 0.0011	0.2687 ± 0.0025	0.0604 ± 0.0096	0.00 ± 0.00
AF-Muon	0.3736 ± 0.0044	6.1092 ± 2.0978	0.9994 ± 0.0001	0.2626 ± 0.0009	0.0396 ± 0.0043	0.00 ± 0.00

worse in this image-token setting, consistent with the tied-table diagnostics: Sign clips all tied-table rows and has much larger tail energy, while AF-Muon keeps the finite-cap geometry active.

The diagnostics confirm that the finite-cap tied-table geometry is active in the image-token setting and behaves as it does in language. Hybrid Muon holds the tied table to a very small norm—its auxiliary AdamW branch applies weight decay—but it uses no clipping geometry, so the row-clipping and tail-energy diagnostics do not apply to it. The two LMO arms differ sharply: SCION-style Sign attains the largest tied-table norm and clips every row, concentrating most of the update energy in the clipped tail, whereas AF-Muon maintains an intermediate norm, clips only a selective fraction of rows, and leaves little energy in the tail. Crucially, AF-Muon's update geometry relative to the Sign endpoint cosine 0.80 and objective ratio 1.25 matches its large-batch language-model values almost exactly, indicating that the same finite-cap tied-table structure governs image tokens and text tokens alike. This shared geometry, together with the norm and tail-energy gap between AF-Muon and the Sign endpoint, explains why AF-Muon is both more stable and better-performing than pure Sign in this cross-domain ablation.

We also evaluate the final checkpoints on a fixed top-half image-completion protocol. For each validation image, the model conditions on the first 512 image tokens and we compute the likelihood of the remaining 512 tokens. This metric does not rely on sampled image quality and provides a paired held-out completion test, where AF-Muon achieves the best score among the trained checkpoints.

Table 20: Tied ImageGPT-style ImageNet-32 results with a deterministic RGB554 16k color vocabulary. Results are mean ± SD over three paired seeds. All methods use the same tied input/output color-token table, FP32 trainable parameters and optimizer state with BF16 autocast, batch size 512, and 1.31B image tokens.

Optimizer	Train loss ↓	Val. loss ↓	Val. bits/dim ↓	Val. PPL ↓
Hybrid Muon	2.3236 ± 0.0067	2.3346 ± 0.0049	1.1227 ± 0.0024	10.33 ± 0.05
SCION-style Sign	2.3568 ± 0.0049	2.3712 ± 0.0055	1.1403 ± 0.0027	10.71 ± 0.06
AF-Muon	**2.3208 ± 0.0101**	**2.3243 ± 0.0055**	**1.1177 ± 0.0026**	**10.22 ± 0.06**

Table 21: Per-seed tied ImageGPT-style ImageNet-32 results. All runs use 1.31B image tokens and the same deterministic RGB554 16k tokenizer. The final row reports a paired two-sided t-test over validation loss between AF-Muon and the strongest non-AF baseline by mean validation loss; the gain is baseline minus AF-Muon, so positive values indicate that AF-Muon is better.

Method	Seed	Train loss ↓	Val. loss ↓	Val. bits/dim ↓	Val. PPL ↓
Hybrid Muon	43	2.3301	2.3363	1.1235	10.34
Hybrid Muon	44	2.3240	2.3290	1.1200	10.27
Hybrid Muon	45	2.3166	2.3384	1.1245	10.36
SCION-style Sign	43	2.3537	2.3671	1.1383	10.67
SCION-style Sign	44	2.3625	2.3690	1.1392	10.69
SCION-style Sign	45	2.3543	2.3775	1.1433	10.78
AF-Muon	43	2.3300	2.3275	1.1193	10.25
AF-Muon	44	2.3100	2.3180	1.1147	10.16
AF-Muon	45	2.3225	2.3273	1.1192	10.25
AF-Muon vs. Hybrid Muon (paired t-test)		validation-loss gain $= 0.0103$, $p = 0.0053$			

Table 22: Final tied-table diagnostics for the tied ImageGPT-style ImageNet-32 ablation. Values are mean ± SD over three paired seeds. The cosine and objective-ratio columns compare the AF-Muon tied-table update geometry to the Sign endpoint.

Method	Tied RMS	Tied max	Rows clipped	Tail energy	Cos(AF, Sign)	Obj. ratio(AF/Sign)
Hybrid Muon	0.0327 ± 0.0000	0.1430 ± 0.0021	–	–	–	–
SCION-style Sign	0.5332 ± 0.0037	4.9483 ± 0.6251	1.000 ± 0.000	0.8012 ± 0.0005	–	–
AF-Muon	0.3096 ± 0.0006	2.2829 ± 0.3463	0.773 ± 0.015	0.0350 ± 0.0013	0.799 ± 0.001	1.251 ± 0.000

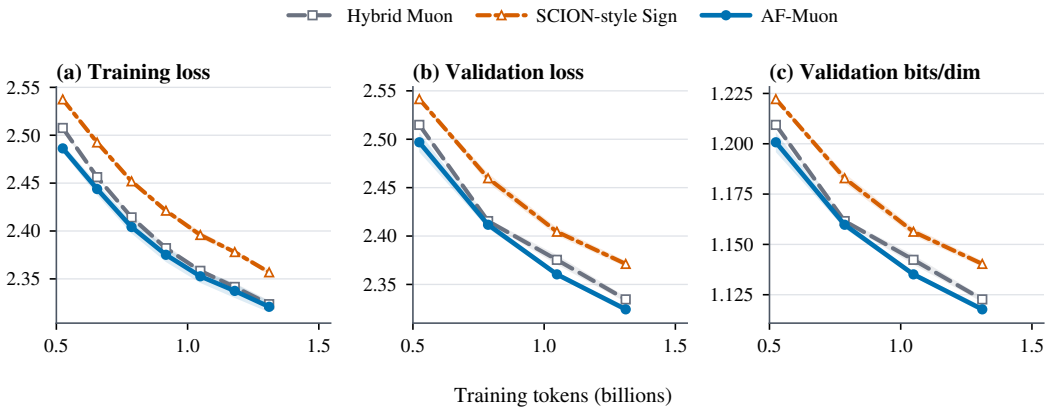


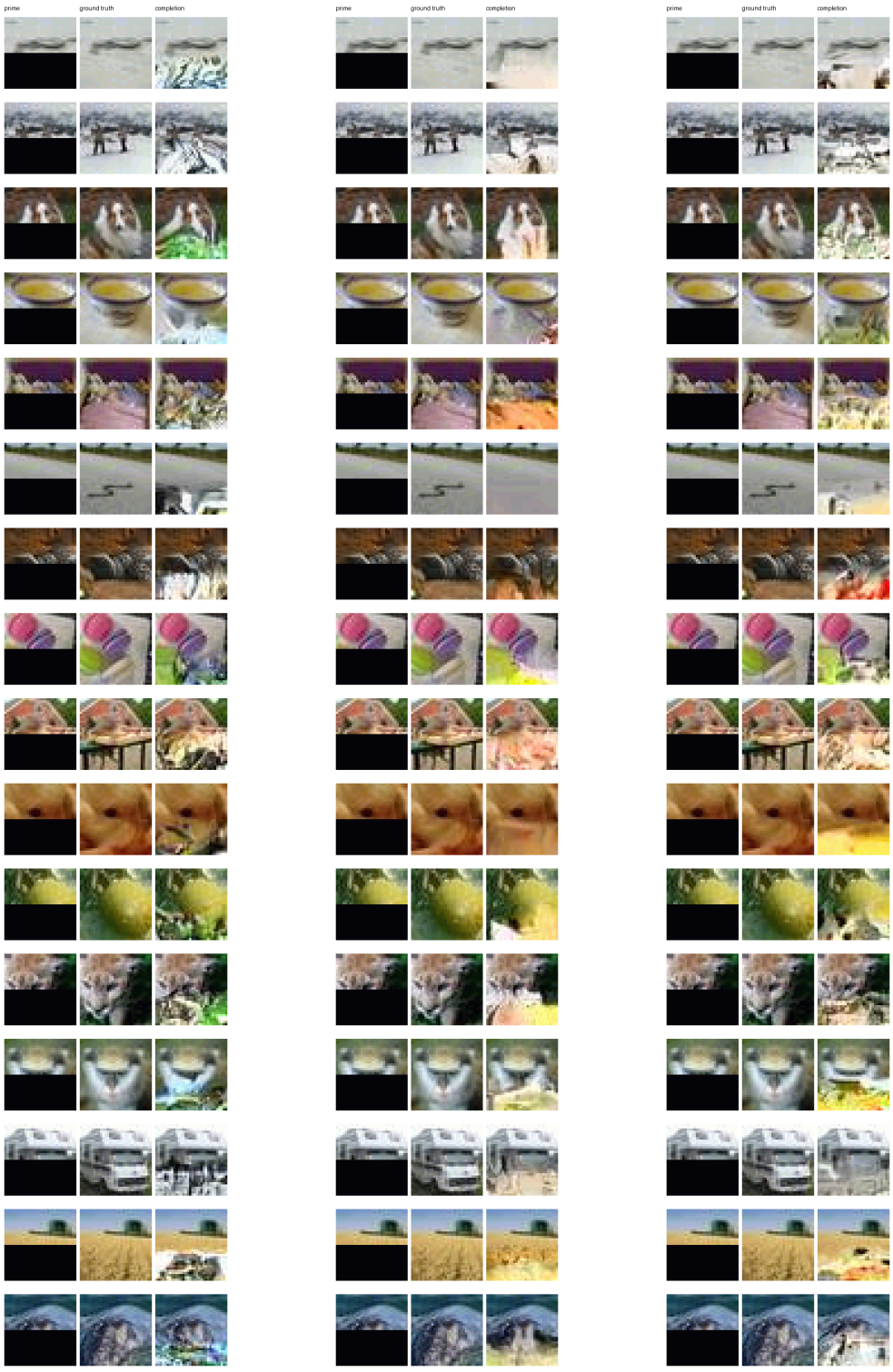
Figure 13: Tied ImageGPT-style ImageNet-32 training dynamics with a deterministic RGB554 16k color vocabulary. Curves show the mean over three paired seeds, and shaded regions denote ±1 standard error of the mean. To focus on the stable late-training regime, the plot starts at step 1000; training loss is shown every 250 steps, and validation metrics are shown every 500 steps. AF-Muon achieves the best final validation loss and validation bits/dim, while SCION-style Sign remains worse throughout this late window.

Table 23: Top-half image-completion metrics for the tied ImageGPT-style ImageNet-32 experiment. Metrics are computed by conditioning on the first 512 tokens of fixed validation images and evaluating the likelihood of the held-out bottom half. Results are mean ± SD over three trained seeds; all methods use the same validation images.

Optimizer	Completion loss ↓	Bits/token ↓	Bits/dim ↓	PPL ↓
Hybrid Muon	2.0273 ± 0.0038	2.9248 ± 0.0054	0.9749 ± 0.0018	7.59 ± 0.03
SCION-style Sign	2.0511 ± 0.0105	2.9591 ± 0.0151	0.9864 ± 0.0050	7.78 ± 0.08
AF-Muon	**2.0117 ± 0.0076**	**2.9023 ± 0.0109**	**0.9674 ± 0.0036**	**7.48 ± 0.06**

Table 24: Per-seed top-half completion results for tied ImageGPT-style ImageNet-32. The same fixed validation images and completion protocol are used for all methods.

Method	Seed	Completion loss ↓	Bits/token ↓	Bits/dim ↓	PPL ↓
Hybrid Muon	43	2.025113	2.921620	0.973973	7.58
Hybrid Muon	44	2.0316	2.9310	0.9770	7.63
Hybrid Muon	45	2.025116	2.921625	0.973975	7.58
SCION-style Sign	43	2.0456	2.9512	0.9837	7.73
SCION-style Sign	44	2.0631	2.9765	0.9922	7.87
SCION-style Sign	45	2.0444	2.9495	0.9832	7.72
AF-Muon	43	2.0205	2.9149	0.9716	7.54
AF-Muon	44	2.007315	2.895943	0.965314	7.44
AF-Muon	45	2.007310	2.895937	0.965312	7.44



(a) Hybrid Muon (b) SCION-style Sign (c) AF-Muon

Figure 14: Qualitative top-half completions from the tied ImageGPT-style ImageNet-32 seed-43 checkpoints. Each panel uses the same fixed validation images and the same sampling seed. Within each panel, columns show the top-half prime, the ground-truth full image, and the sampled bottom-half completion. The samples are illustrative only; quantitative completion likelihoods are reported in Table 23.

J.8 Protein Language Modeling

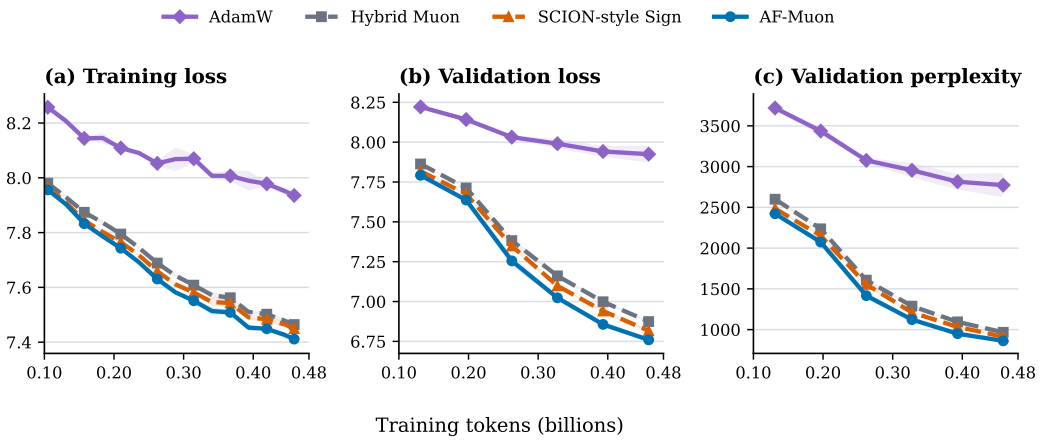


Figure 15: Protein language-modeling training dynamics on UniRef50 / ProtGPT2-BPE with a 363M-parameter tied-vocabulary RoPE decoder. Curves show the mean over three paired seeds under the matched large-batch protocol, after omitting the first 100M training tokens to focus on the stable training regime. AF-Muon maintains the lowest training loss, validation loss, and validation perplexity among the tested optimizers.

We evaluate protein-domain transfer on UniRef50 using a ProtGPT2-BPE tokenizer and a tied-vocabulary 363M-parameter RoPE decoder. All optimizer arms use the same model, data, precision, batch size, and token budget. The matched rerun uses FP32 trainable parameters and optimizer state with BF16 autocast, sequence length 512, micro-batch size 32, gradient accumulation 32, and 524,288 tokens per optimizer update. We compare all methods at 458.8M training tokens over three paired seeds. We tune the AdamW control with a short seed-43 learning-rate pilot (Table 25), select $\eta = 6 \times 10^{-4}$, and then use this setting for the matched three-seed comparison in Table 26.

Table 25: Seed-43 AdamW LR pilot. All runs use the same 363M RoPE decoder and 500 steps / 32.77M tokens. We use $\eta = 6 \times 10^{-4}$ for the longer AdamW comparison in table

AdamW LR	Eval loss ↓	Eval PPL ↓
1×10^{-4}	8.2669	3892.69
3×10^{-4}	8.2007	3643.67
6×10^{-4}	**8.1369**	**3418.32**
1×10^{-3}	8.1672	3523.36

Table 26: Per-seed protein language-modeling results on UniRef50 / ProtGPT2-BPE. All runs use the matched large-batch protocol and are compared at 458.8M training tokens. The final row reports a paired two-sided t-test over eval loss between AF-Muon and the strongest non-AF baseline by mean eval loss; the gain is baseline minus AF-Muon, so positive values indicate that AF-Muon is better.

Optimizer	Seed	Train step	Eval step	Train loss	Eval loss	Eval ppl	Best loss
AdamW	43	850	875	7.9344	8.0297	3070.73	7.8998
AdamW	44	850	875	7.9773	7.8709	2619.91	7.8709
AdamW	45	850	875	7.9421	7.8737	2627.15	7.8737
Hybrid Muon	43	850	875	7.4835	6.8770	969.68	6.8770
Hybrid Muon	44	850	875	7.4868	6.8806	973.21	6.8806
Hybrid Muon	45	850	875	7.4633	6.8650	958.14	6.8650
SCION-style Sign	43	850	875	7.4729	6.8208	916.68	6.8208
SCION-style Sign	44	850	875	7.4727	6.8260	921.48	6.8260
SCION-style Sign	45	850	875	7.4473	6.8023	899.93	6.8023
AF-Muon	43	850	875	**7.4244**	**6.7469**	**851.43**	**6.7469**
AF-Muon	44	850	875	**7.4455**	**6.7814**	**881.26**	**6.7814**
AF-Muon	45	850	875	**7.4077**	**6.7502**	**854.27**	**6.7502**
AF-Muon vs. SCION-style Sign (paired t-test)				eval-loss gain $= 0.0569$, $p = 0.0231$			

AF-Muon gives the lowest validation loss and perplexity for each seed. Averaged over the three seeds, AF-Muon improves validation loss by 0.1147 over Hybrid Muon and by 0.0569 over the SCION-style Sign endpoint. This matched rerun therefore preserves the protein-domain trend under the same large-batch optimizer protocol used for the main tied-token language-model experiments.

J.9 Training Overhead

We measure practical training overhead under the same matched large-batch protocol used in the main experiments. For each model, the three optimizer arms are run sequentially on the same GPU, with evaluation, checkpointing, and diagnostic logging disabled during the timed region. All runs use FP32 trainable parameters and optimizer state with BF16 autocast, 524k tokens per optimizer update, matrix weight decay 0.1, Hybrid Muon auxiliary/tied weight decay 0.01, and zero auxiliary/tied weight decay for SCION-style Sign and AF-Muon. Median step time is computed after discarding the first five steps as warmup, and peak allocated memory is measured with `torch.cuda.max_memory_allocated()` after resetting CUDA peak statistics immediately before the training loop.

The primary memory saving is in optimizer state: SCION-style Sign and AF-Muon remove the AdamW second-moment buffer from the auxiliary/tied branch, reducing optimizer-state memory by roughly 18–22% relative to Hybrid Muon in these models. Median step-time overhead of AF-Muon relative to Hybrid Muon is at most 1.3% across models, and SCION-style Sign is within 0.4%; removing the second-moment buffer from the auxiliary/tied branch therefore costs negligible throughput. The corresponding reduction in single-GPU peak allocated memory is smaller because these large-batch runs are activation- and temporary-buffer dominated. Across models, absolute throughput and peak memory are not directly comparable because the model size, micro-batch size, and GPU differ; the relevant comparison is the within-model difference between optimizer arms. NanoGPT, SmolLM2-135M, and Qwen2.5-0.5B are benchmarked on NVIDIA A100 40GB GPUs, while Llama-3.2-1B is benchmarked on an NVIDIA H200 GPU.

Table 27: Training-overhead benchmark under the matched large-batch protocol. Optimizer state memory is computed analytically from the parameter partition and FP32 optimizer states. "Saved" reports the optimizer-state reduction relative to Hybrid Muon within the same model. Peak allocated memory is measured exactly with `torch.cuda.max_memory_allocated()` after resetting CUDA peak statistics before the training loop. Step time is the median over 30 training steps after discarding the first five steps as warmup. The short-run training loss is included as a sanity check; long-horizon performance is reported in the main experiment tables.

Model	Optimizer	Opt. state	Saved	Peak alloc.	Step (s)	Tok/s	Rel. thr.	Train loss
NanoGPT	Hybrid Muon	0.63 GiB	–	18.603 GiB	6.306	83.1k	1.000×	6.5425
NanoGPT	SCION-style Sign	0.49 GiB	22.2%	18.461 GiB	6.331	82.8k	0.996×	6.2402
NanoGPT	AF-Muon	0.49 GiB	22.2%	18.461 GiB	6.386	82.1k	0.987×	6.1697
SmolLM2-135M	Hybrid Muon	0.55 GiB	–	25.000 GiB	11.858	44.2k	1.000×	6.8592
SmolLM2-135M	SCION-style Sign	0.45 GiB	18.2%	24.895 GiB	11.864	44.2k	1.000×	6.6022
SmolLM2-135M	AF-Muon	0.45 GiB	18.2%	24.895 GiB	11.928	44.0k	0.994×	6.5485
Qwen2.5-0.5B	Hybrid Muon	2.35 GiB	–	23.260 GiB	24.193	21.7k	1.000×	6.8713
Qwen2.5-0.5B	SCION-style Sign	1.84 GiB	21.7%	22.752 GiB	24.245	21.6k	0.998×	6.7913
Qwen2.5-0.5B	AF-Muon	1.84 GiB	21.7%	22.752 GiB	24.459	21.4k	0.989×	6.6253
Llama-3.2-1B	Hybrid Muon	5.58 GiB	–	74.766 GiB	16.330	32.1k	1.000×	6.7290
Llama-3.2-1B	SCION-style Sign	4.60 GiB	17.6%	73.787 GiB	16.395	32.0k	0.996×	6.7475
Llama-3.2-1B	AF-Muon	4.60 GiB	17.6%	73.787 GiB	16.486	31.8k	0.990×	6.7018

K Ablations

K.1 NanoGPT Factorial Ablation: Tied Table Versus 1D Auxiliary Parameters

A potential concern is that AF-Muon changes both the tied vocabulary table and the remaining one-dimensional auxiliary parameters relative to Hybrid Muon. Therefore, the improvement over Hybrid Muon might be caused by replacing the one-dimensional AdamW auxiliary branch rather than by the proposed finite-cap tied-table geometry. We test this with a 2×2 factorial ablation on NanoGPT.

All runs use the same NanoGPT architecture and training protocol as in the main NanoGPT experiment: FP32 trainable parameters and optimizer state with BF16 autocast, constant learning rates, matrix weight decay 0.1, maximum gradient norm 1.0, and the same FineWeb token stream. The matrix parameters are always optimized with Muon. We independently choose the optimizer used for the tied vocabulary table and the optimizer used for the remaining one-dimensional parameters:

$$\{\text{AdamW tied},\ \text{AF-Muon tied } (c = 3, s = 0.5)\} \times \{\text{AdamW 1D},\ \text{RMS-LMO 1D}\}.$$

All four factorial arms use zero auxiliary weight decay, so the comparison isolates update geometry rather than adding an extra shrinkage term. (Figure 4)

Table 28: NanoGPT 2×2 factorial ablation separating the tied vocabulary-table update from the remaining one-dimensional auxiliary update. All runs use 1.05B training tokens and three paired seeds.

Tied table	1D aux.	Val. loss ↓	Val. PPL ↓
AdamW	AdamW	3.6296 ± 0.0030	37.70 ± 0.11
AF-Muon	AdamW	3.6163 ± 0.0108	37.20 ± 0.40
AdamW	RMS-LMO	3.6221 ± 0.0028	37.41 ± 0.11
AF-Muon	RMS-LMO	$\mathbf{3.5883 \pm 0.0016}$	$\mathbf{36.17 \pm 0.06}$

The isolated tied-table replacement improves validation loss by 0.0133 over the AdamW/AdamW baseline, while the isolated one-dimensional RMS-LMO replacement improves validation loss by 0.0075. Replacing both gives a substantially larger 0.0413 improvement. This shows that the full AF-Muon result is not merely a side effect of changing the one-dimensional auxiliary optimizer. The tied-table finite-cap geometry contributes directly, and the best result appears when it is combined with RMS-LMO updates for the remaining auxiliary parameters.

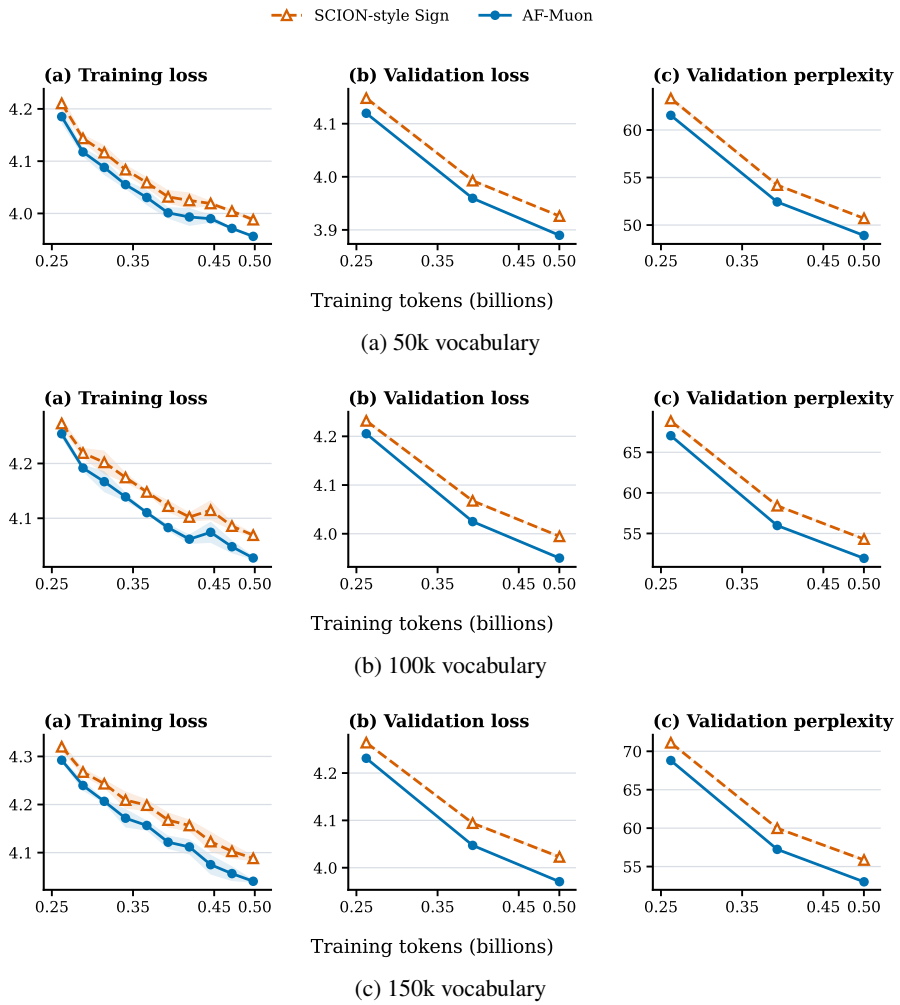
K.2 Real Vocabulary-Size Re-tokenization Ablation

This experiment tests whether vocabulary size alone explains why AF-Muon and the Sign endpoint are closer in some large-vocabulary FP32 runs. We train matched ByteLevel BPE tokenizers with 50k, 100k, and 150k vocabulary sizes on the same FineWeb shard and re-tokenize the data for each vocabulary. This is a real re-tokenization test rather than a padded-vocabulary test: token IDs reach the top of each vocabulary in every setting.

Because different tokenizers induce different per-token entropy scales, absolute losses should not be compared across vocabulary sizes. The relevant comparison is within each vocabulary. Across all three vocabularies, AF-Muon improves over the SCION-style Sign endpoint: by 0.036 validation-loss points at 50k, 0.045 at 100k, and 0.052 at 150k. Thus the AF-Muon–Sign gap does not shrink as the real vocabulary grows in this controlled NanoGPT setting.

Table 29: NanoGPT real-vocabulary re-tokenization ablation. We train matched ByteLevel BPE tokenizers with fix d=768, and 50k, 100k, and 150k vocabulary sizes on the same FineWeb shard, then train each optimizer for 500M tokens. All runs use FP32 trainable parameters and optimizer state with BF16 autocast, matrix weight decay 0.1, zero auxiliary/tied weight decay, and effective batch size 512. The 50k runs use microbatch 16; 100k and 150k use microbatch 8 for memory. We report mean ± sample standard deviation over three paired seeds. related figure 16

Vocab	Optimizer	Params	Train loss ↓	Val. loss ↓	Val. PPL ↓
50k	SCION-style Sign	123.4M	3.9881 ± 0.0079	3.9260 ± 0.0014	50.70 ± 0.07
	AF-Muon	123.4M	**3.9561 ± 0.0017**	**3.8897 ± 0.0051**	**48.90 ± 0.25**
100k	SCION-style Sign	161.8M	4.0691 ± 0.0069	3.9947 ± 0.0041	54.31 ± 0.22
	AF-Muon	161.8M	**4.0275 ± 0.0059**	**3.9500 ± 0.0061**	**51.94 ± 0.32**
150k	SCION-style Sign	200.2M	4.0881 ± 0.0108	4.0228 ± 0.0070	55.86 ± 0.39
	AF-Muon	200.2M	**4.0405 ± 0.0055**	**3.9706 ± 0.0041**	**53.02 ± 0.22**



(a) 50k vocabulary

(b) 100k vocabulary

(c) 150k vocabulary

Figure 16: Real-vocabulary re-tokenization ablation on NanoGPT. Each row uses a separately trained ByteLevel BPE tokenizer with vocabulary size 50k, 100k, or 150k and compares AF-Muon with the SCION-style Sign endpoint for 500M training tokens. Curves show the mean over three paired seeds, with shaded regions denoting ±1 standard error. Since tokenizers differ across rows, absolute validation losses and perplexities should not be compared across vocabulary sizes; the relevant comparison is the within-vocabulary AF-Muon versus Sign gap.

K.3 TIED-STEP MAGNITUDE ABLATION

Table 30 reports the NanoGPT tied-scale sweep. The model architecture, vocabulary, data, precision recipe, matrix weight decay, and token budget are fixed across all rows. The only AF-Muon hyperparameter varied is the tied-table scale s, with c fixed to 3.

Across the tested values, AF-Muon obtains lower validation loss and perplexity than the SCION-style Sign endpoint. The best validation loss in this sweep is obtained at $s = 0.25$, while nearby values $s = 0.375$ and $s = 0.1875$ give similar results. The tied-table RMS decreases as s is reduced, reflecting the smaller tied-table update scale.

Table 30: NanoGPT AF-Muon tied-scale sweep at fixed model width and vocabulary size. All runs use a 12-layer NanoGPT model with hidden size $d = 768$, a 50k vocabulary, 500M training tokens, FP32 trainable parameters and optimizer state with BF16 autocast, matrix weight decay 0.1, and zero auxiliary/tied weight decay. AF-Muon uses $c = 3$ and varies the tied scale s. The SCION-style Sign endpoint uses $c = 1, s = 1$ under the same training protocol.

Method	s	d_{eff}	Train loss ↓	Val. loss ↓	Val. PPL ↓	Tied RMS
SCION-style Sign	1.0000	–	3.9971	3.9250	50.65	0.2147
AF-Muon	0.5000	768	3.9558	3.8867	48.75	0.1241
AF-Muon	0.4300	893	3.9510	3.8799	48.42	0.1066
AF-Muon	0.3750	1024	3.9409	3.8697	47.93	0.0956
AF-Muon	0.2500	1536	3.9377	3.8687	47.88	0.0692
AF-Muon	0.1875	2048	3.9406	3.8720	48.04	0.0554
AF-Muon	0.1250	3072	3.9522	3.8856	48.70	0.0444

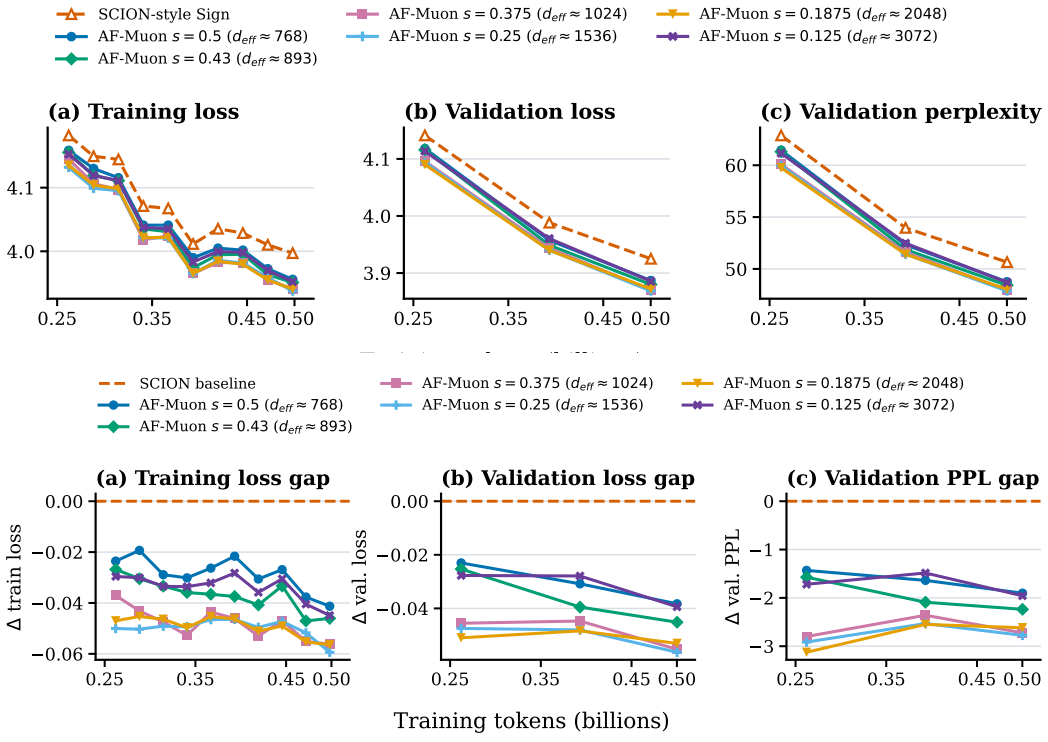


Figure 17: NanoGPT tied-scale sweep training dynamics. Top: training loss, validation loss, and validation perplexity for SCION-style Sign and AF-Muon with different tied scales s. Bottom: token-wise AF-Muon minus SCION-style Sign gaps for the same metrics. All runs use the same fixed 50k vocabulary, $d = 768$ model width, 500M training-token budget, FP32 trainable parameters and optimizer state with BF16 autocast, matrix weight decay 0.1, and zero auxiliary/tied weight decay.

K.4 NanoGPT Real-Width Ablation

We next test whether the AF-Muon advantage over the Sign endpoint persists when the tied vocabulary rows have larger real hidden dimension. Unlike the tied-scale diagnostic, this ablation changes the actual model width and tied-row dimension. All runs use a 50k ByteLevel BPE tokenizer, 500M training tokens, FP32 trainable parameters and optimizer state with BF16 autocast, matrix weight decay 0.1, zero tied/auxiliary weight decay for the LMO methods, and three paired seeds. AF-Muon uses $c = 3, s = 0.5$; the SCION-style Sign endpoint uses $c = 1, s = 1$.

Table 31: Real-width NanoGPT ablation at fixed 50k vocabulary and 500M training tokens. We report mean $\pm$ sample standard deviation over three paired seeds. Rows clipped denotes the fraction of tied-table rows for which the finite-cap constraint is active in the final tied-momentum diagnostic.

Width d	Optimizer	Params	Train loss ↓	Val. loss ↓	Val. PPL ↓
768	SCION-style Sign	123.4M	3.9881 ± 0.0079	3.9260 ± 0.0014	50.70 ± 0.07
768	AF-Muon	123.4M	$\mathbf{3.9561 \pm 0.0017}$	$\mathbf{3.8897 \pm 0.0051}$	$\mathbf{48.90 \pm 0.25}$
1024	SCION-style Sign	202.2M	3.9205 ± 0.0035	3.8464 ± 0.0005	46.82 ± 0.02
1024	AF-Muon	202.2M	$\mathbf{3.8822 \pm 0.0026}$	$\mathbf{3.8068 \pm 0.0026}$	$\mathbf{45.00 \pm 0.12}$
1536	SCION-style Sign	416.6M	3.8475 ± 0.0020	3.7573 ± 0.0041	42.83 ± 0.18
1536	AF-Muon	416.6M	$\mathbf{3.8136 \pm 0.0031}$	$\mathbf{3.7253 \pm 0.0051}$	$\mathbf{41.48 \pm 0.21}$

AF-Muon remains better than the Sign endpoint at all three real widths. The paired AF-Muon minus Sign validation-loss gaps are -0.0363 ± 0.0063 at $d = 768$, -0.0397 ± 0.0023 at $d = 1024$, and -0.0321 ± 0.0085 at $d = 1536$. Thus increasing real hidden width attenuates neither the sign nor the consistency of the finite-cap advantage in this NanoGPT setting, although the row-clipping diagnostic moves closer to the Sign endpoint as width increases.

Table 32: Per-seed real-width NanoGPT results for the width ablation in Table 31.

Width d	Optimizer	Seed	Train loss ↓	Val. loss ↓	Val. PPL ↓
768	SCION-style Sign	43	3.9971	3.9250	50.65
768	SCION-style Sign	44	3.9844	3.9253	50.67
768	SCION-style Sign	45	3.9827	3.9276	50.79
768	AF-Muon	43	3.9558	3.8867	48.75
768	AF-Muon	44	3.9546	3.8956	49.19
768	AF-Muon	45	3.9580	3.8869	48.76
1024	SCION-style Sign	43	3.9235	3.8465	46.83
1024	SCION-style Sign	44	3.9214	3.8469	46.85
1024	SCION-style Sign	45	3.9166	3.8459	46.80
1024	AF-Muon	43	3.8850	3.8092	45.11
1024	AF-Muon	44	3.8818	3.8071	45.02
1024	AF-Muon	45	3.8799	3.8040	44.88
1536	SCION-style Sign	43	3.8487	3.7555	42.75
1536	SCION-style Sign	44	3.8451	3.7544	42.71
1536	SCION-style Sign	45	3.8485	3.7621	43.04
1536	AF-Muon	43	3.8168	3.7308	41.71
1536	AF-Muon	44	3.8106	3.7243	41.44
1536	AF-Muon	45	3.8134	3.7207	41.29

Table 33: Paired AF-Muon minus SCION-style Sign gaps for the real-width NanoGPT ablation. Negative values indicate that AF-Muon is better.

Width d	Seeds	Δ Val. loss	Δ Val. PPL
768	43,44,45	-0.0363 ± 0.0058	-1.81 ± 0.28
1024	43,44,45	-0.0397 ± 0.0023	-1.82 ± 0.10
1536	43,44,45	-0.0321 ± 0.0085	-1.35 ± 0.36

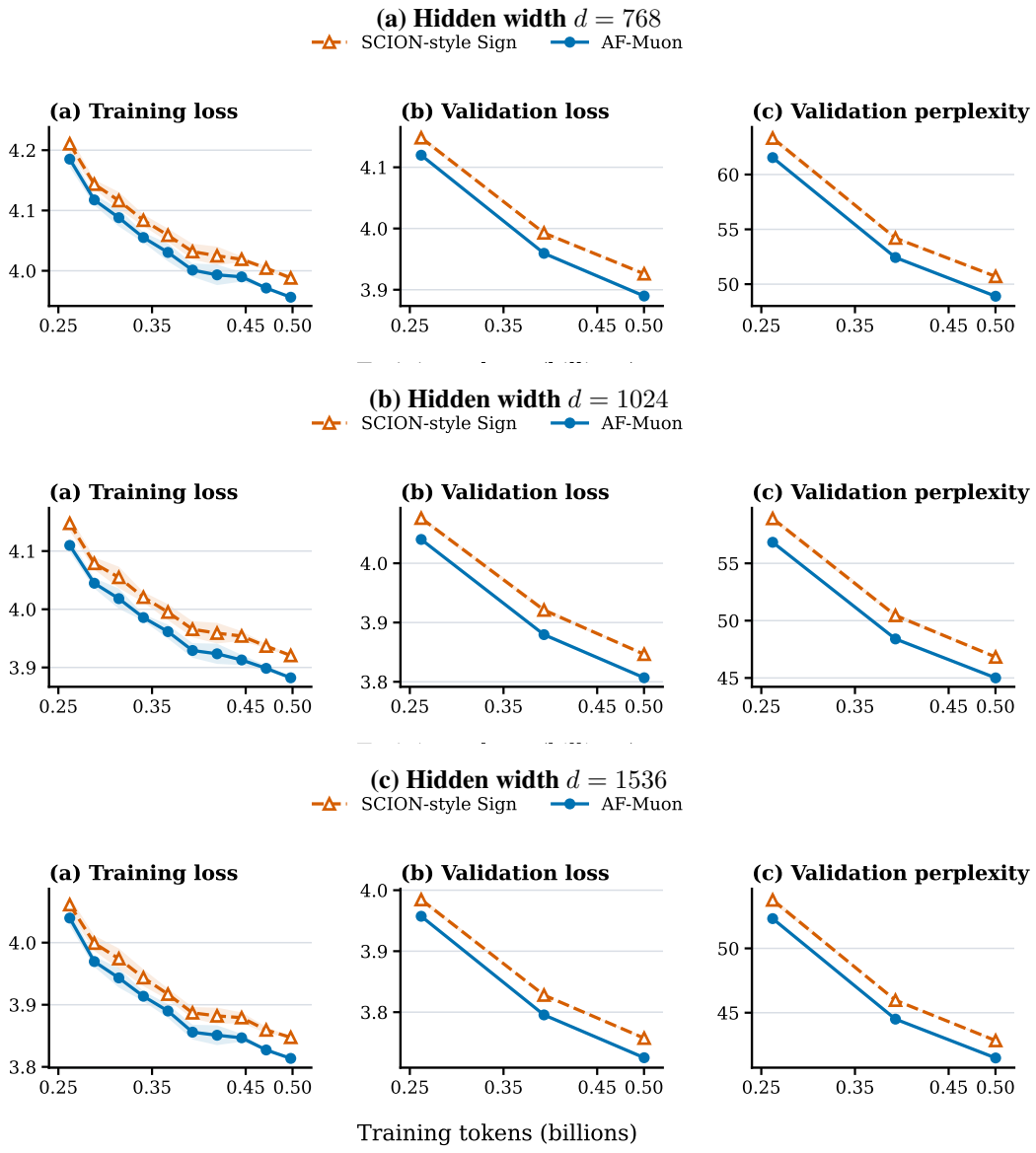


Figure 18: Real-width NanoGPT training dynamics at fixed 50k vocabulary and 500M training tokens. Each row compares AF-Muon and the SCION-style Sign endpoint for one hidden width $d \in \{768, 1024, 1536\}$. Curves show the mean over three paired seeds, with shaded regions denoting ± 1 standard error.

K.5 Batch-size Sensitivity

We next test whether the AF-Muon tied-table advantage depends on the particular large-batch setting used in the main NanoGPT experiment. We use the same NanoGPT-style tied-embedding model, FineWeb tokenization, optimizer hyperparameters, FP32 trainable parameters and optimizer state with BF16 autocast, matrix learning rate 0.02, auxiliary/vector learning rate 3×10^{-4}, matrix weight decay 0.1, and gradient clipping at norm 1.0. We vary only the global batch size, keeping the micro-batch size fixed at 16 and changing gradient accumulation accordingly. All methods are compared at the same token budget of 458.8M tokens.

Table 34: NanoGPT batch-size sensitivity at a matched 458.8M-token comparison point. All runs use seed 43. Tokens/update is the number of training tokens per optimizer step.

Batch	Tokens/update	Hybrid Muon	SCION-style Sign	AF-Muon	AF–Sign
32	32,768	4.2078	4.1472	**4.1378**	-0.0094
64	65,536	3.9918	3.9738	**3.9514**	-0.0224
128	131,072	3.8571	3.8559	**3.8180**	-0.0379
256	262,144	3.7950	3.7890	**3.7544**	-0.0346
512	524,288	3.8200	3.7649	**3.7400**	-0.0249

AF-Muon gives the lowest validation loss at every tested batch size, from 32k to 524k tokens per optimizer update. The advantage over the Sign endpoint is smallest at the smallest batch and becomes clearer for the intermediate and large-batch settings. This indicates that the finite-cap tied-table update is not narrowly tuned to the default large-batch protocol: it remains beneficial across a broad range of effective batch sizes under the same matrix-update and precision settings.

Figure 5 visualizes the same matched-token comparison. AF-Muon remains below both baselines across the full tested range of global batch sizes, indicating that the finite-cap tied-table update is not narrowly tuned to a single effective batch size.

K.6 Auxiliary Learning-rate Sensitivity

Auxiliary/fallback learning-rate sensitivity. Throughout the main experiments, we use the standard Muon parameter grouping: hidden matrices are updated by Muon with learning rate 0.02, while parameters outside the hidden matrix group, such as embeddings, classifier heads, normalization scales, and biases, are assigned to an auxiliary AdamW-style branch with learning rate 3×10^{-4} in the reference Muon recipe (Jordan et al., 2024). The public Muon implementation also describes this learning rate as being in spectral-norm units for hidden matrices and recommends AdamW for the non-hidden parameters. We apply this grouping convention across models, replacing the AdamW treatment of the tied vocabulary table by the SCION-style Sign endpoint or by AF-Muon where appropriate, while keeping the hidden-matrix Muon learning rate fixed.

To test whether the NanoGPT comparison depends on a single auxiliary scale, Table 35 sweeps the auxiliary/fallback learning rate around and beyond the Muon default. We use the same NanoGPT-style tied-embedding model and FineWeb tokenization as in Appendix J.1. All runs use seed 43, FP32 trainable parameters and optimizer state with BF16 autocast, matrix learning rate 0.02, matrix weight decay 0.1, gradient clipping at norm 1.0, batch size 512, micro-batch size 16, and 524,288 tokens per optimizer update. We vary the auxiliary/fallback learning rate over

$$7.5 \times 10^{-5},\ 1.5 \times 10^{-4},\ 3 \times 10^{-4},\ 6 \times 10^{-4},\ 1.2 \times 10^{-3},\ 2.4 \times 10^{-3},\ 9.6 \times 10^{-3},$$

corresponding to $0.25\times$ through $32\times$ the default 3×10^{-4}. For Hybrid Muon, this branch includes the tied vocabulary table; for the AdamW-free tied-table arms, the tied table is updated by the Sign or AF-Muon LMO and the sweep tests the remaining auxiliary/vector scale. All methods are compared at the same 458.8M-token point validation loss.

The sweep shows that Hybrid Muon is sensitive to the auxiliary AdamW branch: increasing this learning rate substantially narrows the gap relative to AF-Muon. However, AF-Muon remains competitive across the extended range and attains the best loss at most tested settings before the

Table 35: NanoGPT auxiliary/fallback learning-rate sensitivity at a matched 458.8M-token comparison point. All runs use seed 43. The hidden-matrix Muon learning rate is fixed at 0.02; the auxiliary/fallback learning-rate scale is varied. For Hybrid Muon this branch includes the tied vocabulary table, while the AdamW-free arms use their tied-table LMO and vary the remaining auxiliary/vector scale.

Aux/tied LR	Mult.	Hybrid Muon	SCION-style Sign	AF-Muon	AF–Sign	AF–Hybrid
7.5×10^{-5}	$0.25\times$	4.1498	3.7743	**3.7557**	-0.0185	-0.3941
1.5×10^{-4}	$0.5\times$	3.9716	3.7664	**3.7472**	-0.0192	-0.2244
3.0×10^{-4}	$1.0\times$	3.8182	3.7726	**3.7337**	-0.0389	-0.0845
6.0×10^{-4}	$2.0\times$	3.7641	3.7522	**3.7284**	-0.0238	-0.0357
1.2×10^{-3}	$4.0\times$	3.7304	3.7393	**3.7220**	-0.0173	-0.0084
2.4×10^{-3}	$8.0\times$	3.7233	3.7283	**3.7208**	-0.0075	-0.0024
9.6×10^{-3}	$32.0\times$	3.7393	**3.7371**	3.7372	0.0001	-0.0021

largest rates degrade all methods. We therefore view this as a sensitivity diagnostic under literature-motivated Muon defaults, not as a claim of compute-optimal tuning for every optimizer.

K.7 AF-Muon Cap and Scale Sensitivity

We evaluate sensitivity to the tied-table coordinate cap c and update scale s using the NanoGPT-style tied-vocabulary model and large-batch protocol of Appendix J.1. We test $c \in \{1, 2, 3, 4, 6, 10, \infty\}$ and $s \in \{0.5, 1\}$ with three matched seeds per configuration, for a total of 42 training runs. All configurations use the same NanoGPT tied-vocabulary task and are evaluated at optimizer step 750, corresponding to 393.2M training tokens. The endpoints $c = 1$ and $c = \infty$ recover Sign and row-RMS, respectively.

Table 36: AF-Muon cap and scale sensitivity on the NanoGPT tied-vocabulary task at the 393.2M-token intermediate comparison point. Results are mean $\pm$ SD over three paired seeds. The default setting used in the main experiments is $c = 3$, $s = 0.5$.

Cap c	Scale s	Train loss	Val. loss	Val. PPL	Tied RMS	Tied max
1	0.5	3.7880 ± 0.0208	3.7831 ± 0.0033	43.95 ± 0.15	0.1270 ± 0.0008	0.6653 ± 0.0033
2	0.5	3.7885 ± 0.0250	3.7826 ± 0.0021	43.93 ± 0.09	0.1444 ± 0.0004	1.1725 ± 0.0324
3	0.5	$\mathbf{3.7872 \pm 0.0253}$	$\mathbf{3.7815 \pm 0.0026}$	$\mathbf{43.88 \pm 0.12}$	0.1418 ± 0.0029	1.5712 ± 0.0923
4	0.5	3.7892 ± 0.0241	3.7856 ± 0.0062	44.06 ± 0.27	0.1409 ± 0.0039	1.7930 ± 0.1915
6	0.5	3.7910 ± 0.0232	3.7845 ± 0.0005	44.01 ± 0.02	0.1408 ± 0.0017	2.1051 ± 0.2569
10	0.5	3.7909 ± 0.0244	3.7867 ± 0.0014	44.11 ± 0.06	0.1399 ± 0.0018	2.5313 ± 0.2197
∞	0.5	3.7919 ± 0.0227	3.7844 ± 0.0010	44.01 ± 0.05	0.1404 ± 0.0019	2.6954 ± 0.0666
1	1	3.8168 ± 0.0301	3.8109 ± 0.0064	45.19 ± 0.29	0.2407 ± 0.0039	1.2568 ± 0.0039
2	1	3.8250 ± 0.0243	3.8214 ± 0.0042	45.67 ± 0.19	0.2778 ± 0.0043	2.3705 ± 0.0609
3	1	3.8288 ± 0.0221	3.8124 ± 0.0065	45.26 ± 0.30	0.2755 ± 0.0045	3.1119 ± 0.2910
4	1	3.8291 ± 0.0289	3.8252 ± 0.0049	45.84 ± 0.23	0.2701 ± 0.0011	4.0566 ± 0.0744
6	1	3.8285 ± 0.0213	3.8214 ± 0.0071	45.67 ± 0.32	0.2692 ± 0.0029	4.8737 ± 0.7576
10	1	3.8199 ± 0.0220	3.8160 ± 0.0042	45.42 ± 0.19	0.2688 ± 0.0049	5.6160 ± 0.2148
∞	1	3.8244 ± 0.0186	3.8171 ± 0.0092	45.47 ± 0.42	0.2695 ± 0.0035	7.6845 ± 0.1437

Update scale has the larger effect in this sweep. Reducing s from 1 to 0.5 lowers mean validation loss for every tested cap. At $s = 0.5$, the mean losses across caps lie in a narrow band. Thus, calibrating the tied-table update magnitude has a larger effect than selecting the precise cap within the tested grid. In particular, comparisons between the default $(c, s) = (3, 0.5)$ and the standard Sign endpoint $(1, 1)$ reflect changes in both geometry and scale.

The finite-cap default remains competitive at matched scale. The default $(c, s) = (3, 0.5)$ achieves the lowest mean validation loss in this intermediate sweep: 3.7815 ± 0.0026, compared with 3.7831 ± 0.0033 for scale-matched Sign and 3.7844 ± 0.0010 for scale-matched row-RMS. These margins are small, so we do not interpret the observed ordering as establishing statistical superiority.

Table 37: Tied-table diagnostics for the NanoGPT cap and scale sweep at 393.2M training tokens. Results are mean ± SD over three matched seeds. Coordinate and row fractions describe cap activity; tail energy is computed from the row-RMS endpoint before applying the finite-cap oracle. Cosine similarity and linear-objective ratio compare the normalized AF-Muon direction with Sign evaluated on the same oracle input within each run.

Cap c	Scale s	Matrix RMS	Coord. capped	Rows clipped	Tail energy	Cos(AF, Sign)	Obj. ratio
1	0.5	0.0493 ± 0.0002	1.0000 ± 0.0000	1.0000 ± 0.0000	0.8005 ± 0.0002	1.000 ± 0.000	1.000 ± 0.000
2	0.5	0.0501 ± 0.0001	0.0579 ± 0.0000	1.0000 ± 0.0000	0.2723 ± 0.0021	0.818 ± 0.000	1.246 ± 0.000
3	0.5	0.0500 ± 0.0001	0.0028 ± 0.0001	0.8949 ± 0.0042	0.0303 ± 0.0014	0.799 ± 0.000	1.253 ± 0.000
4	0.5	0.0500 ± 0.0002	0.0001 ± 0.0000	0.0484 ± 0.0050	0.0012 ± 0.0002	0.798 ± 0.000	1.253 ± 0.000
6	0.5	0.0500 ± 0.0001	0.0000 ± 0.0000	0.0002 ± 0.0004	0.0000 ± 0.0000	0.798 ± 0.000	1.253 ± 0.000
10	0.5	0.0499 ± 0.0001	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.798 ± 0.000	1.253 ± 0.000
∞	0.5	0.0500 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.798 ± 0.000	1.253 ± 0.000
1	1	0.0557 ± 0.0003	1.0000 ± 0.0000	1.0000 ± 0.0000	0.7988 ± 0.0009	1.000 ± 0.000	1.000 ± 0.000
2	1	0.0589 ± 0.0002	0.0588 ± 0.0003	1.0000 ± 0.0000	0.3024 ± 0.0077	0.817 ± 0.000	1.251 ± 0.002
3	1	0.0594 ± 0.0007	0.0032 ± 0.0003	0.9195 ± 0.0180	0.0450 ± 0.0133	0.799 ± 0.001	1.255 ± 0.002
4	1	0.0594 ± 0.0002	0.0004 ± 0.0001	0.2476 ± 0.0625	0.0196 ± 0.0093	0.797 ± 0.001	1.257 ± 0.002
6	1	0.0593 ± 0.0002	0.0000 ± 0.0000	0.0238 ± 0.0384	0.0016 ± 0.0027	0.797 ± 0.001	1.255 ± 0.002
10	1	0.0593 ± 0.0003	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.797 ± 0.000	1.254 ± 0.000
∞	1	0.0596 ± 0.0002	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.797 ± 0.000	1.254 ± 0.000

At $s = 1$, Sign has the lowest mean loss, further indicating that no cap is uniformly best across tested scales.

Selective coordinate control motivates the default. At the same 393.2M-token audit, $c = 3, s = 0.5$ caps only 0.0028 ± 0.0001 of coordinates while affecting 0.8949 ± 0.0042 of vocabulary rows. The cap therefore acts on a small fraction of coordinates but across most rows. Its normalized direction attains a same-state linear-objective ratio of 1.253 ± 0.000 relative to Sign. By construction, the normalized direction satisfies $\|u\|_\infty \le c$; for rows where the row constraint is active, this also bounds coordinate concentration. These properties distinguish the finite-cap update from Sign, which assigns equal magnitude to nonzero coordinates, and from row-RMS, which lacks the additional coordinate constraint.

Row-RMS tail-energy diagnostic. To characterize the coordinate concentration encountered by the cap, we compute the stabilized row-RMS endpoint

$$R_i(B) = \frac{B_i}{\max\{\mathrm{RMS}(B_i), \varepsilon\}}$$

and define its tail-energy fraction at threshold c as

$$E_{\mathrm{tail}}(B; c) = \frac{\sum_{i,j} R_{ij}(B)^2 \mathbf{1}\{|R_{ij}(B)| > c\}}{\sum_{i,j} R_{ij}(B)^2},$$

with value zero when the denominator is zero. This diagnostic measures the fraction of squared row-RMS endpoint energy carried by coordinates exceeding the proposed cap. It is evaluated before applying the finite-cap oracle and does not measure energy removed, linear-objective retention, or validation performance. At threshold $c = 1$, approximately 80% of the row-RMS endpoint energy lies in coordinates exceeding the threshold. This describes the pre-oracle row-RMS direction; the completed Sign direction has equal magnitudes on its nonzero coordinates. For the default $(c, s) = (3, 0.5)$, $(3.03 \pm 0.14)\%$ of the squared row-RMS endpoint energy lies in coordinates whose absolute values exceed 3.

Together, these results support $c = 3, s = 0.5$ as a practical default with selective coordinate control and competitive validation performance. They also limit the claim: the sweep supports the combined scale-calibrated AF-Muon recipe, but does not establish a universal performance advantage for the finite cap over appropriately scaled endpoints.

K.8 Sparse-input and Dense-output Gradient Sources

A tied input/output vocabulary table receives two structurally different gradient sources: sparse input-lookup gradients from the embedding use, and dense output-classifier gradients from the language-

modeling head. We isolate these sources by running AF-Muon on the same NanoGPT-style tied-embedding model while replacing the tied-table gradient before the optimizer step. The three variants use only the sparse input component, only the dense output component, or the normal sum of both components. All other parameters, including the Transformer matrices and one-dimensional auxiliary parameters, are trained with the standard AF-Muon recipe. All runs use seed 43, FP32 trainable parameters and optimizer state with BF16 autocast, matrix learning rate 0.02, auxiliary/vector learning rate 3×10^{-4}, matrix weight decay 0.1, zero auxiliary/tied weight decay, gradient clipping at norm 1.0, batch size 512, micro-batch size 16, and 524,288 tokens per optimizer update. We compare all variants at 458.8M training tokens.

Table 38: NanoGPT tied-table gradient-source ablation at a matched 458.8M-token comparison point. Only the gradient used for the tied input/output vocabulary table is changed; all other model parameters use the same AF-Muon update.

Tied-table gradient	Train loss	Val. loss	Val. PPL	Input rows	Output rows	Input-output cos.
Sparse input only	4.0353	4.0513	57.47	0.601	–	–
Dense output only	3.7471	3.7634	43.10	–	1.000	–
Sparse input + dense output	**3.7217**	**3.7395**	**42.08**	0.601	1.000	0.003

The dense output-classifier gradient is the dominant tied-table signal: removing it and using only sparse input-lookup gradients substantially worsens validation loss. However, the sparse input signal is not redundant. Using only the dense output component reaches validation loss 3.7634, while the full tied gradient improves to 3.7395. The input/output gradient cosine is near zero, showing that the sparse lookup component supplies a distinct direction rather than a scaled copy of the dense classifier gradient.

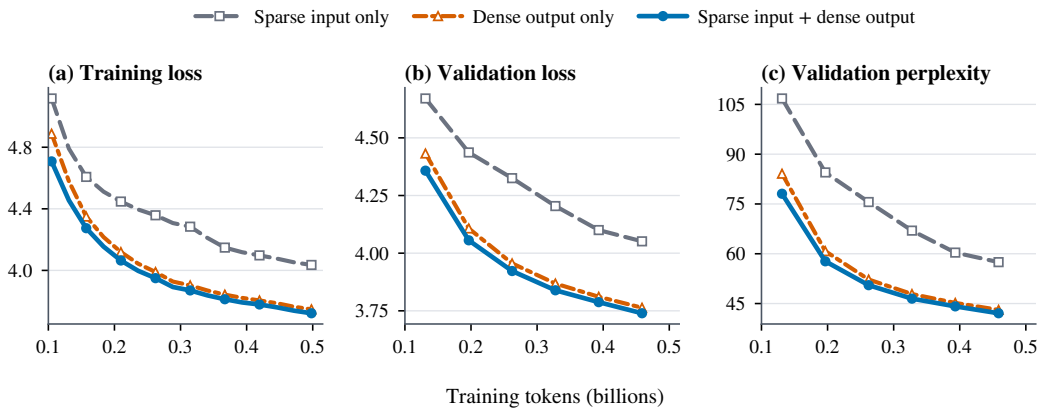


Figure 19: Training dynamics for the NanoGPT tied-table gradient-source ablation. Curves show seed-43 runs after the first 100M training tokens. All variants use the same AF-Muon optimizer for non-tied parameters; only the gradient supplied to the tied input/output vocabulary table is changed. The dense output-only variant learns much faster than the sparse input-only variant, but the full sparse-input plus dense-output tied gradient remains best.

Figure 19 shows that the advantage of the full tied gradient persists throughout training: the sparse input-only variant remains far behind, while the full sparse-plus-dense update consistently improves over the dense output-only update.

K.9 Boundary case: homogeneous dense-parameter reuse

AF-Muon is motivated by a parameter tensor that is shared across *heterogeneous* functional roles. This raises a natural boundary question: does every form of parameter sharing require an intersection geometry, or is ordinary Muon already sufficient when the shared uses have the same operator type?

To examine this distinction, we construct a small diagnostic in which the same MLP up- and down-projection matrices are reused at four successive depths. This setting produces several gradient contributions for one stored parameter, but unlike a tied embedding/output table, every use remains a dense linear map with the same spectral geometry.

Shared-use gradient decomposition. Let a matrix W be used at L different depths. Introduce conceptual copies $W^{(1)}, \dots, W^{(L)}$ and write the loss before imposing parameter sharing as

$$\widetilde{\mathcal{L}}\left(W^{(1)}, \dots, W^{(L)}\right).$$

The shared model is obtained by imposing

$$W^{(1)} = \cdots = W^{(L)} = W.$$

The gradient on the stored shared parameter is therefore

$$G := \nabla_W \mathcal{L} = \sum_{\ell=1}^{L} G_\ell, \qquad G_\ell := \nabla_{W^{(\ell)}} \widetilde{\mathcal{L}}\Big|_{W^{(1)}=\cdots=W^{(L)}=W}. \tag{25}$$

Thus, parameter sharing causes autograd to accumulate the use-specific gradients G_ℓ into one stored gradient G.

The important distinction is that the individual G_ℓ may differ because each depth sees different activations and residual states, but all uses of W still represent the same type of dense linear operator. Consequently, they induce the same spectral constraint.

Why the intersection collapses. Let the feasible update set for each dense use be the same spectral ball

$$\mathcal{C}_{\mathrm{sp}} = \{U : \|U\|_{\mathrm{op}} \le r\}.$$

If every occurrence of the shared matrix has this geometry, then intersecting the role constraints produces no new feasible set:

$$\bigcap_{\ell=1}^{L} \mathcal{C}_{\mathrm{sp}} = \mathcal{C}_{\mathrm{sp}}. \tag{26}$$

Moreover, because one update U is applied to every occurrence of W, its first-order effect is

$$\sum_{\ell=1}^{L} \langle G_\ell, U\rangle = \left\langle \sum_{\ell=1}^{L} G_\ell, U \right\rangle = \langle G, U\rangle.$$

The correct shared-parameter LMO is therefore

$$D_{\mathrm{sp}}(G) \in \arg\max_{U \in \mathcal{C}_{\mathrm{sp}}} \left\langle \sum_{\ell=1}^{L} G_\ell, U \right\rangle. \tag{27}$$

In the exact spectral case, $D_{\mathrm{sp}}(G)$ is the polar factor of the accumulated gradient G. In the implementation, it is approximated by the same Newton–Schulz map used by Muon.

This is different from independently computing one Muon direction per use and then summing, $V_{\mathrm{raw}} = \sum_{\ell=1}^{L} D_{\mathrm{sp}}(G_\ell)$. Because the LMO is nonlinear, $D_{\mathrm{sp}}\left(\sum_{\ell=1}^{L} G_\ell\right) \neq \sum_{\ell=1}^{L} D_{\mathrm{sp}}(G_\ell)$ in general. Furthermore, V_{raw} generally lies outside the single shared spectral ball because its operator norm can grow with L. Consequently, the raw difference between these two expressions is not a fair same-radius comparison.

We instead rescale the summed per-use direction to the spectral radius of the aggregate Muon direction:

$$\widetilde{V} = \frac{\|D_{\mathrm{sp}}(G)\|_{\mathrm{op}}}{\|V_{\mathrm{raw}}\|_{\mathrm{op}}} V_{\mathrm{raw}}. \tag{28}$$

We then measure its equal-radius linear-objective retention:

$$R_{\mathrm{obj}} = \frac{\langle G, \widetilde{V}\rangle}{\langle G, D_{\mathrm{sp}}(G)\rangle}. \tag{29}$$

For an exact LMO and exact operator norms, $R_{\mathrm{obj}} \le 1$. A value below one measures the linearized objective lost by applying the nonlinear spectral map separately to each use before combining the directions.

Table 39: **Homogeneous dense-sharing diagnostic.** One up-projection and one down-projection are reused at four depths. Statistics are measured at the final audit after 5.0M predicted tokens. Objective retention compares the equal-spectral-radius sum of per-depth Muon directions with Muon applied once to the accumulated gradient. The experiment uses one seed and is intended as a boundary diagnostic rather than a performance comparison.

Matrix	Mean pair cosine	Min. pair cosine	Negative pairs	Norm CV	Cancellation κ	Direction cosine	Objective retained	Split error
Up projection	0.3396	0.0507	0.0%	0.3247	0.6803	0.6785	74.08%	5.22×10^{-8}
Down projection	0.5014	0.2251	0.0%	0.2800	0.7697	0.7764	83.85%	5.11×10^{-8}

Diagnostic setup. We use a four-depth causal diagnostic model with hidden size 512, feed-forward size 2048, and a vocabulary of 49,152 tokens. The same MLP up-projection and down-projection matrices are reused at every depth, while the normalization parameters remain depth-specific. The input embedding, positional embedding, and output head are untied so that the experiment does not contain the tied-vocabulary mechanism studied in the main paper.

The model contains 52.6M parameters and is trained on 5.0M predicted tokens from FineWeb. We use FP32 parameters and optimizer state, no autocast, Muon learning rate 0.02, auxiliary learning rate 3×10^{-4}, matrix weight decay 0.1, and auxiliary AdamW weight decay 0.01. Only the shared dense matrices are relevant to this diagnostic. Because Hybrid Muon and AF-Muon use the same Muon branch for these matrices, this is a diagnostic of the common hidden-matrix update rather than a comparison between complete optimizers.

For the final audit batch, we replace the shared matrix by L equal-valued differentiable copies and compute the use-specific gradients $G_1, \ldots, G_L$. We validate the decomposition by comparing $\sum_\ell G_\ell$ with the gradient obtained from the genuinely shared forward pass.

We report the pairwise depth-gradient cosine

$$C_{\ell k} = \frac{\langle G_\ell, G_k \rangle}{\|G_\ell\|_F \|G_k\|_F},$$

the coefficient of variation of the depth-gradient norms, and the cancellation ratio

$$\kappa = \frac{\left\| \sum_{\ell=1}^{L} G_\ell \right\|_F}{\sum_{\ell=1}^{L} \|G_\ell\|_F}. \tag{30}$$

Here $\kappa = 1$ indicates perfectly aligned contributions, whereas smaller values indicate increasing cancellation or directional diversity.

Results. The split-gradient reconstruction is numerically exact to floating-point precision. The shared and split computations have identical loss and logits, and the relative gradient reconstruction errors are 5.22×10^{-8} and 5.11×10^{-8} for the up and down matrices, respectively. No nonfinite values are observed.

The use-specific gradients are not identical. Their mean pairwise cosines are 0.3396 for the up projection and 0.5014 for the down projection, and their norm coefficients of variation are 0.3247 and 0.2800. Nevertheless, every pairwise cosine remains positive: the minimum cosines are 0.0507 and 0.2251, and the fraction of negative depth pairs is zero for both matrices. The cancellation ratios are 0.6803 and 0.7697, indicating moderate directional diversity but no severe destructive conflict.

Muon applied to the accumulated gradient differs substantially from the raw sum of per-depth Muon directions, with direction cosines 0.6785 and 0.7764. This difference is expected from the nonlinearity of the spectral LMO and is not evidence of an optimizer failure. After matching spectral radius, the per-depth-then-sum construction retains only 74.08% of the aggregate Muon linear objective for the up projection and 83.85% for the down projection. Thus, applying Muon once to the accumulated gradient is materially better aligned with the shared parameter's first-order objective than applying Muon independently to each occurrence and subsequently combining the directions.

Training itself remains stable throughout the diagnostic. Validation loss decreases from approximately 10.91 at initialization to 6.35 after 5.0M tokens, the final reported gradient norm is 0.71, and no nonfinite parameters, gradients, or optimizer states are observed. We do not interpret the absolute

loss as a competitive language-modeling result because the model is an intentionally restricted shared-MLP diagnostic without attention or cross-token mixing.

Boundary with tied-table sharing. This experiment identifies an important boundary for role-aware optimizer design. Repeated parameter use is not by itself sufficient to require a new LMO. In the shared dense-layer setting, different use sites generate different gradients, but every use induces the same spectral update geometry. The intersection of their feasible sets therefore collapses to the original spectral ball, and the standard Muon rule applied to the accumulated gradient is the appropriate shared-parameter LMO.

The tied embedding/output table studied in the main paper lies on the other side of this boundary. Its input uses act as vocabulary-indexed lookup rows, whereas its output use acts as a dense vocabulary classifier. These roles motivate distinct row-wise and entrywise controls:

$$\mathcal{C}_{\mathrm{tie},c} = \left\{ U : \max_i \frac{\|u_i\|_2}{\sqrt{d}} \le 1, \quad \max_{i,j} \frac{|U_{ij}|}{c} \le 1 \right\}.$$

Their intersection does not collapse and produces the finite-cap clipped-row LMO. The relevant distinction is therefore not merely whether a tensor is aliased, but whether its aliases impose heterogeneous update geometries:

$$\boxed{\text{homogeneous reuse} \implies \text{accumulate gradients and apply the existing LMO,}}$$

whereas

$$\boxed{\text{heterogeneous role sharing} \implies \text{construct a joint feasible geometry.}}$$

The dense-sharing diagnostic therefore serves as a negative boundary control: it shows why AF-Muon's new tied-table geometry is specialized to heterogeneous lookup/classifier aliasing rather than being a generic modification for every shared parameter.