OptiSelect: How does the Optimizer Shape Data Curriculum?

Anonymous authors
Paper under double-blind review

Abstract

Online data selection has demonstrated substantial efficiency gains for LLM pretraining by training on the most valuable candidates within each batch. Since a candidate's value is realized through its effective model update, principled selection should account for the optimizer step, which reshapes the raw gradient before it updates model parameters. We formalize this optimizer-aware selection paradigm as *OptiSelect* and present the first systematic study of how the optimizer shapes data selection. Our theory establishes a *selection gain principle* in which the advantage of online selection is governed by the *discriminability* of the optimizer-induced utility scores. We prove that sign-based and polar-tangential preconditioners of `Lion` and `Muon` would suffer from a *discriminability collapse* which caps attainable gains from *OptiSelect*, whereas diagonal-adaptive optimizers such as `AdamW` and `Sophia` admit strictly better upper bounds. The proposed principle also yields a quantitative derivation of the optimal candidate oversampling ratio. Pretraining experiments on 124M and 720M models are consistent with our theoretical analysis and show that `AdamW`'s diagonal-adaptive scoring geometry remains the strongest scoring geometry even with `Muon` as optimizer. We further demonstrate that *OptiSelect* retains its benefits under data rephrasing, a technique used in modern data processing pipelines. Our findings provide theoretical foundations and practical guidance for co-designing optimizers and data selection in LLM pretraining.

1 Introduction

Data recipes and optimizers are two important levers determining the efficiency of LLM pretraining: they define what the model learns from and how it learns, respectively. These two design choices are often considered independently. Data curation pipelines filter or reweight the corpus before training begins, agnostic to the optimizer that will process it (Xie et al., 2023a; Fan et al., 2024; Li et al., 2024; Penedo et al., 2024; Wang et al., 2024), while optimizer benchmarks compare optimization algorithms on uniformly sampled data streams, agnostic to data selection strategies (Semenov et al., 2025; Wen et al., 2026). The *optimizer-aware online data selection* strategy (Wang et al., 2026) combines these two considerations, which we formalize as the *OptiSelect* training paradigm: At every optimization step, each data sample from an oversampled candidate pool is scored by its marginal utility, and only the highest-utility batch is selected for training. As a sample's utility is realized through the model update it induces, the data utilities should be computed with the optimizer preconditioner, which reshapes the raw gradient before updating model parameters. Therefore, the optimizer plays a dual role under *OptiSelect*. It descends the loss landscape strategically while also designing the data curriculum, adapting the data utility scoring function according to its unique preconditioner.

We analyze the interaction between optimizers and data selection across five optimizers spanning different preconditioning geometries: `AdamW` rescales gradient coordinates by their estimated RMS (Loshchilov & Hutter, 2019), `AdEMAMix` mixes in a slow momentum buffer (Pagliardini et al., 2025), `Sophia` normalizes by estimated curvature (Liu et al., 2024), `Lion` discards magnitudes and keeps only signs (Chen et al., 2023), and `Muon` orthogonalizes each update matrix (Jordan, 2024). Our study is organized around one central question from both theoretical and empirical perspectives:

How does the optimizer interact with online data selection,
and which optimization geometries can actually profit from it?

Contributions. We summarize our contributions as three-fold.

• **The *OptiSelect* framework (§3).** We formalize the *OptiSelect* training paradigm through an optimizer-aware *transformation operator* $\mathcal{O}_t$ and derive the corresponding optimizer-aware data utility functions across five optimizers spanning various geometries.

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.

• **A principled theory of the optimizer–selection interplay (§4).** We establish a *selection gain principle*, demonstrating that the one-step advantage from data selection scales with the *discriminability* of the utility scores (Theorem 1). We prove that a *sign collapse* bounds the gain of `Lion` and a *radial collapse* bounds that of `D-Muon`, whereas `AdamW` and `Sophia` retain strictly larger score spread. The same principle also derives the optimum candidate oversampling ratio as $\lambda^\star = 2$.

• **A comprehensive pretraining benchmark (§5).** We evaluate five optimizers under standard training and *OptiSelect* on 124M and 720M models, demonstrating a close agreement between measured gains and the theory. Decoupling the scoring geometry from the base optimizer reveals that the optimizer-matched geometry is not necessarily optimal, with `AdamW`'s diagonal-adaptive operator prevailing under `D-Muon`. The selection benefit persists under format rephrasing.

2 Related Work

We position each work briefly in this section and provide a more detailed survey in Appendix A.

Data selection for LLM pretraining. Research on training-data ordering and sample weighting predates LLMs, e.g. easy-to-hard curricula (Bengio et al., 2009), online loss-ranked sampling (Loshchilov & Hutter, 2015), importance sampling with learned gradient-norm proxies (Katharopoulos & Fleuret, 2018) whereas none of these strategies explicitly accounts for the optimizer. At LLM scale, selection is dominated by *static* pipelines that curate the corpus before training begins: hashed n-gram resampling (DSIR; Xie et al., 2023b), fastText quality classifiers (DCLM; Li et al., 2024), distilled LLM quality judgments (FineWeb-Edu; Penedo et al., 2024), and proxy-model domain weights (DoReMi; Xie et al., 2023a, DoGE; Fan et al., 2024). The online strategies include MATES (Yu et al., 2024), which predicts influence with a periodically refreshed auxiliary model; GREATS (Wang et al., 2024) and OPUS (Wang et al., 2026), which compute gradient-alignment scores and optimizer-aware scores, respectively.

Optimizer-aware influence. Influence functions (Koh & Liang, 2017), the first-order influence TracIn (Pruthi et al., 2020) and TRAK with random projections (Park et al., 2023) attributed model behaviour to training samples under implicit `SGD` dynamics. Adaptive optimizer state changes influence: LESS (Xia et al., 2024) and TrackStar (Chang et al., 2025) correct influence scores for `Adam`'s second moment; OPUS (Wang et al., 2026) computes optimizer-aware attribution scores on `AdamW` and `Muon`, and uses them for per-step sample selection during LLM pretraining. Yet, none compares the influence geometries different optimizers induce, which is the object of our theory (§4).

Optimizers for LLM pretraining. The families we study span diagonal-adaptive (`AdamW`; `AdEMAMix`; Pagliardini et al., 2025), curvature-aware (`Sophia`; Liu et al., 2024), sign-based (`Lion`, `Signum`), and matrix-geometry methods (`Muon`; Jordan, 2024), with `SOAP` (Vyas et al., 2025), `MARS` (Yuan et al., 2025). Systematic benchmarks evaluate these under standard pretraining only (Semenov et al., 2025; Wen et al., 2026), where rankings already shift with scale and batch size. No prior work explores how optimizer algorithms affect preferences in *data selection*, which we analyze both theoretically (§4) and empirically (§5) in this work.

Algorithm 1 *OptiSelect*: one pretraining step with optimizer-aware data selection

Require: Candidate batch $\tilde{\mathcal{B}}_t$, parameters $\boldsymbol{\theta}_t$, optimizer state, target batch size B, temperature τ
Ensure: Updated parameters $\boldsymbol{\theta}_{t+1}$
// Candidate forward-backward pass
for each $z \in \tilde{\mathcal{B}}_t$ **do**
  Compute per-sample gradient $g_z \leftarrow \nabla_{\boldsymbol{\theta}} \ell(z; \boldsymbol{\theta}_t)$
end for
// Influence scoring
for each $z \in \tilde{\mathcal{B}}_t$ **do**
  Assign utility $U(z, t)$ using g_z and the current optimizer state
end for
// Selection via Boltzmann distribution
Define $p_t(z) \propto \exp\big(U(z,t)/\tau\big)$ for $z \in \tilde{\mathcal{B}}_t$
Sample $\mathcal{B}_t \subset \tilde{\mathcal{B}}_t$ with $|\mathcal{B}_t| = B$ according to p_t
// Optimizer update
$\boldsymbol{\theta}_{t+1} \leftarrow \textsc{OptimizerStep}(\boldsymbol{\theta}_t, \{g_z\}_{z \in \mathcal{B}_t})$
return $\boldsymbol{\theta}_{t+1}$

3 The *OptiSelect* Paradigm

In this section, we first formalize the *OptiSelect* training loop (§3.1), then detail the optimizer-induced preconditioning that defines the data attribution score for online selection (§3.2).

3.1 Data Schemes in LLM Pretraining

Standard pretraining. Let $\mathcal{D}$ denote the pretraining data corpus. Standard pretraining minimizes the next-token prediction loss for language modeling, i.e. $\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(z; \boldsymbol{\theta})]$, where $\ell(z; \boldsymbol{\theta})$ is the cross-entropy loss on data instance z. At each step t, a mini-batch $\mathcal{B}_t \subset \mathcal{D}$ of size B is drawn uniformly at random and used to compute a stochastic gradient $\hat{\mathbf{g}}_t = \frac{1}{B} \sum_{z \in \mathcal{B}_t} \nabla_{\boldsymbol{\theta}} \ell(z; \boldsymbol{\theta}_t)$. The model

parameters are updated as $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \cdot \mathrm{Opt}(\hat{\mathbf{g}}_t, \mathcal{S}_t)$, where Opt is the optimizer's step function corresponding to the current optimizer state $\mathcal{S}_t$, e.g. momentum buffers and variance estimates.

***OptiSelect*: Online in-batch data selection.** While uniform sampling provides no control over data quality, *OptiSelect* applies *online data selection* at each optimization step. Specifically, denoting the effective training batch size by B, a larger *candidate batch* $\tilde{\mathcal{B}}_t$ of size $|\tilde{\mathcal{B}}_t| = \lambda B$ is drawn at every step t. Each datapoint is ranked by the optimizer-aware influence score $U(z,t)$; the top-B subset $\mathcal{B}_t \subset \tilde{\mathcal{B}}_t$ is then selected for the model update. The full procedure is presented in Algorithm 1. Under *OptiSelect* the optimizer plays a two-fold role: it performs the optimization step on the loss landscape, and simultaneously shapes data attribution by defining $U(z,t)$. We detail the optimizer-transformed utility under various instantiations in the following section.

3.2 Optimizer-Aware Influence Scoring

Preliminaries and notation. We begin by establishing a unified notation for all following derivations. At step t, we denote $\boldsymbol{\theta}_t \in \mathbb{R}^d$ as the model parameters, the per-sample loss on document z as $\ell(z;\boldsymbol{\theta}_t)$ and its gradient as $\mathbf{g}_{z,t} \triangleq \nabla_{\boldsymbol{\theta}} \ell(z;\boldsymbol{\theta}_t)$ and the stochastic mini-batch gradient $\hat{\mathbf{g}}_t = \frac{1}{|\mathcal{B}_t|}\sum_{z\in\mathcal{B}_t} \mathbf{g}_{z,t}$. We denote element-wise multiplication, division and square-root operations by $\odot$, $\oslash$, and $\sqrt{\cdot}$ respectively. We define the frozen-state scoring operator $\mathcal{O}_t$ as the map from raw gradients to directions used for per-sample attribution, conditional on the optimizer state $\mathcal{S}_t$. Given a candidate z, its scoring direction is $u_{z,t} = \mathcal{O}_t(g_{z,t})$. This operator need not coincide with the actual optimizer update map $\mathrm{Opt}(\cdot, \mathcal{S}_t)$; the relation between them, including the approximations used for scoring, is specified in Appendix B.

Optimizer-transformed Data Influence. During online data selection, we assess the influence of a candidate sample z at step t using its *per-sample scoring direction* $\mathbf{u}_{z,t}$ induced by the optimizer state $\mathcal{S}_t$ and a proxy set from a predefined target distribution. Let $\mathcal{V}$ be a small held-out validation set representative of a target distribution. We define the validation loss $F(\boldsymbol{\theta}) = \frac{1}{|\mathcal{V}|}\sum_{z\in\mathcal{V}} \ell(z;\boldsymbol{\theta})$. Selection proceeds with greedy influence scoring at every step: writing $B^{(t)}$ as the selected batch at time step t, the parameter vector predicted by the additive scoring model is $\boldsymbol{\theta}_t^{\mathrm{sel}} \triangleq \boldsymbol{\theta}_t - \eta_t \sum_{z'\in\mathcal{S}^{(t)}} \mathcal{O}_t(\mathbf{g}_{z',t})$. We estimate the *marginal utility* of adding sample z into $B^{(t)}$ as the additional reduction in F: $U(z,t) = F\big(\boldsymbol{\theta}_t^{\mathrm{sel}}\big) - F\big(\boldsymbol{\theta}_t^{\mathrm{sel}} - \eta_t\, \mathcal{O}_t(\mathbf{g}_{z,t})\big)$. Expanding F to second order around $\boldsymbol{\theta}_t$, denoting $\mathbf{g}_{\mathcal{V},t} = \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}_t)$ as the validation gradient proxy and $\mathbf{H}_F = \nabla^2 F(\boldsymbol{\theta}_t)$ as the validation Hessian, the marginal utility is defined as $U(z,t) = \eta_t \langle \mathcal{O}_t(\mathbf{g}_{z,t}),\, \mathbf{g}_{\mathcal{V},t}\rangle - \eta_t^2\, |B^{(t)}| \langle \mathcal{O}_t(\mathbf{g}_{z,t}),\, \mathbf{H}_F\, \mathbf{G}_t\rangle + \mathcal{O}(\eta_t^2\, \|\mathcal{O}_t(\mathbf{g}_{z,t})\|^2)$. Following OPUS (Wang et al., 2026), we approximate the intractable $d \times d$ Hessian by the identity matrix $\mathbf{H}_F \approx \mathbf{I}$ in the redundancy term , giving

$$U(z,t) \;\approx\; \eta_t\, \langle \mathcal{O}_t(\mathbf{g}_{z,t}),\, \mathbf{g}_{\mathcal{V},t}\rangle \;-\; \eta_t^2\, |B^{(t)}| \langle \mathcal{O}_t(\mathbf{g}_{z,t}),\, \mathbf{G}_t\rangle, \tag{1}$$

where $\mathbf{G}_t = \frac{1}{|B^{(t)}|}\sum_{z'\in\mathcal{S}^{(t)}} \mathcal{O}_t(\mathbf{g}_{z',t})$ is the average scoring direction of the selected samples. The first term measures *alignment* between sample z and the target distribution, while the second penalizes *redundancy* among the peer candidates in the same batch.

Instantiation of $\mathcal{O}_t$ across various optimizer families. Each optimizer algorithm induces a specific operator $\mathcal{O}_t$ for online data attribution at step t. We write $\mathbf{x}$ for the *generic argument* of $\mathcal{O}_t$, i.e. the per-sample raw gradient vector $\mathbf{x} = \mathbf{g}_{z,t} \in \mathbb{R}^d$ or the mini-batch gradient $\mathbf{x} = \hat{\mathbf{g}}_t \in \mathbb{R}^d$, where $\mathbf{d}_t = \mathcal{O}_t(\hat{\mathbf{g}}_t)$ denotes the batch direction represented by the scoring map. For `Muon` the argument is matrix-valued, i.e. $\mathbf{G} \in \mathbb{R}^{m\times n}$, acting on each 2-D weight block in the transformer model. The remaining symbols are read from the optimizer state $\mathcal{S}_t$, which is identical for all candidates within one scoring pass at step t. We present the transformation operators for the benchmarked optimizers, with detailed derivations provided in Appendix B:

$$\begin{aligned}
\mathcal{O}_t^{\texttt{AdamW}}(\mathbf{x}) &= \mathcal{O}_t^{\texttt{AdEMAMix}}(\mathbf{x}) = \mathbf{c}_t \odot \mathbf{x}, & \mathcal{O}_t^{\texttt{Sophia}}(\mathbf{x}) &= \mathbf{x} \oslash \max\{\gamma\, \mathbf{h}_t,\, \varepsilon\},\\
\mathcal{O}_t^{\texttt{Lion}}(\mathbf{x}) &= \mathrm{sign}\big(\beta\, \mathbf{m}_{t-1} + (1-\beta)\, \mathbf{x}\big), & \mathcal{O}_t^{\texttt{Lion, lin}}(\mathbf{x}) &= \mathrm{sign}(\mathbf{m}_{t-1}) \odot \mathbf{x},\\
\mathcal{O}_t^{\texttt{Muon,lin}}(\mathbf{G}) &= \mathrm{D}\,\mathrm{msign}(\mathbf{A}_t)[\mathbf{G}], & \mathcal{O}_t^{\texttt{Muon,rp}}(\mathbf{G}) &= \mathbf{G}\,(\mathbf{A}_t^\top \mathbf{A}_t + \varepsilon \mathbf{I})^{-1/2},
\end{aligned} \tag{2}$$

Specifically, for `AdamW`/ `AdEMAMix` (Appendix B.2, B.3), $\mathbf{c}_t = \mathbf{1} \oslash (\sqrt{\hat{\mathbf{v}}_t} + \varepsilon)$ denotes the inverse root of the bias-corrected second moment $\hat{\mathbf{v}}_t$. `AdEMAMix` induces the *same* operator; for `Sophia` (Appendix B.4), $\mathbf{h}_t$ denotes the estimate of diagonal curvature with trust scale γ; for `Lion` (Appendix B.5), $\mathbf{m}_{t-1}$ denotes the sign methods' momentum buffer, with decay β. With $\beta \gg 1-\beta$; for

Muon (Appendix B.7), $\mathbf{A}_t$ denotes the momentum matrix. $\mathrm{D}\,\mathrm{msign}(\mathbf{A}_t)[\cdot]$ is the faithful Fréchet derivative in $\mathcal{O}_t^{\mathtt{Muon,lin}}$, while $\mathcal{O}_t^{\mathtt{Muon,rp}}$ is a cheaper right-preconditioned proxy.

4 What Makes an Optimizer Good at Selecting Data?

Since each optimizer shapes the influence function through its own transformation $\mathcal{O}_t$, a natural question arises: *how does the choice of optimizer interact with data selection?* To study this question, we introduce ***discriminability*** (Theorem 1), a principled criterion that determines an upper bound on the gain of *OptiSelect* over standard training under a given optimizer. We then analyze several optimization geometries through the lens of this *selection gain principle*: we find that sign collapse and radial collapse cap the achievable selection gain for the sign-based optimizer Lion and the polar-tangential optimizer D-Muon, respectively, whereas diagonal-adaptive optimizers such as AdamW and Sophia enjoy strictly better upper bounds on the selection gain. Our proof builds on a Gaussian model of the data candidates. We detail the setup as follows.

Setup and Notations. At step t, the candidate batch $\tilde{\mathcal{B}}_t$ of size λB is drawn i.i.d. from the data distribution with utility scores follows a Gaussian distribution $\{U(z,t)\}_{z\in\tilde{\mathcal{B}}_t} \sim \mathcal{N}(\mu_U(t), \sigma_U^2(t))$. Consider that candidate gradients are also sampled from a Gaussian distribution $\mathbf{g}_z \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with diagonal covariance $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$. Given the validation proxy $\mathbf{g}_{\mathcal{V}}$, we denote $\hat{\mathbf{g}}_{\mathcal{V}} = \mathbf{g}_{\mathcal{V}} - \eta_t G_t$ to absorb the redundancy term in (1). Denote $\boldsymbol{\pi}$ as the diagonal operator applied by $\mathcal{O}_t$ transformation, the utility score on the sample z can be written as $U_z = \eta_t \langle \boldsymbol{\pi} \odot \mathbf{g}_z, \hat{\mathbf{g}}_{\mathcal{V}} \rangle = \eta_t \langle \mathbf{g}_z, \boldsymbol{\pi} \odot \hat{\mathbf{g}}_{\mathcal{V}} \rangle$, i.e. $\boldsymbol{\pi} = \mathbf{1}$ for SGD, $\boldsymbol{\pi} = \mathbf{c}_t$ (2) for AdamW, and $\boldsymbol{\pi} = \mathrm{sign}(\mathbf{m}_{t-1})$ for linearized sign-based optimizers (Proposition 1).

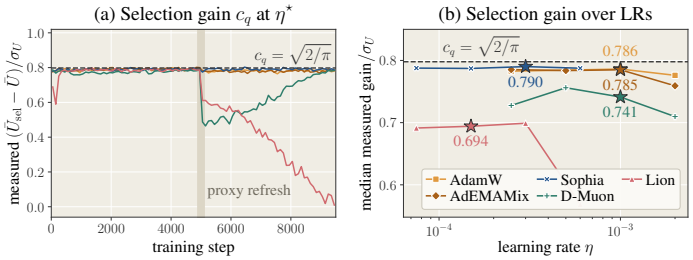


Figure 1: ***Selection gain* (Theorem 1) measured during 124M LLM pretraining.** (a) **Running medians at tuned $\eta^\star$**: the measured gain mostly sits close to c_q for *every* optimizer; while Lion decays after proxy refresh. (b) **Measured gain from learning rate sweep**: the selection gain is flat in various learning rate η.

Theorem 1 (Selection-gain Principle)**.** *Consider the candidate utilities distribution $\mathcal{N}(\mu_U(t), \sigma_U^2(t))$ and standardized top-q superquantile $c_q > 0$[1], where $q = 1/\lambda$ is the retained fraction from OptiSelect. The expected one-step reduction on the validation loss F, $R_t \triangleq \mathbb{E}[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}_{t+1})]$, satisfies*

$$R_t^{\mathrm{rand}} = \mu_U(t) + \mathrm{O}(\eta_t^2), \qquad R_t^{\mathrm{sel}} = \mu_U(t) + c_q\,\sigma_U(t) + \mathrm{O}(\eta_t^2), \tag{3}$$

*which gives the **selection gain**, i.e. the extra loss reduction obtained from data selection as*

$$G_t \triangleq R_t^{\mathrm{sel}} - R_t^{\mathrm{rand}} = c_q\,\sigma_U(t) + \mathrm{O}(\eta_t^2). \tag{4}$$

The proof is given in Appendix C. Since c_q depends only on the retained fraction $q = 1/\lambda$, the absolute gain is governed by the score spread $\sigma_U(t)$. We validate Theorem 1 on 15 *OptiSelect* LLM pretraining runs and compute $(\bar{U}_{\mathrm{sel}} - \bar{U})/\sigma_U$ on the attribution scores, where the proxy gradient is refreshed at step $5k$. Figure 1 shows that the normalized score gain $(\bar{U}_{\mathrm{sel}} - \bar{U})/\sigma_U$ remains consistently close to c_q and nearly invariant across various learning rates. Since c_q depends only on the retained fraction $q = 1/\lambda$, the absolute gain is governed by the score spread $\sigma_U(t)$. We further decompose the variance by coordinates and define the coordinate weights on SGD as $w_j \triangleq \dfrac{\sigma_j^2[\hat{\mathbf{g}}_{\mathcal{V}}]_j^2}{\sum_k \sigma_k^2[\hat{\mathbf{g}}_{\mathcal{V}}]_k^2}$. In the rest of this section, we adopt the *variance ratio* $\dfrac{\mathrm{Var}[U_z^{(\boldsymbol{\pi})}]}{\mathrm{Var}[U_z^{\mathrm{SGD}}]} = \sum_{j=1}^{d} w_j \pi_j^2$ to compare the *discriminability* benefits of linear operators against SGD.

Sign Collapse. Sign-based optimizers can lose *discriminability* since the momentum buffer easily dominates the sign of gradients, diminishing the candidate-aware differences. The selector therefore

[1] After standardizing the candidate utilities to zero mean and unit variance, the *standardized top-q superquantile* (c_q) refers to the mean utility of the top $q = 1/\lambda$ candidates. For Gaussian utilities at $\lambda = 2$, c_q is the mean of the upper half of a standard normal, $\sqrt{2/\pi} \approx 0.798$.

has little information for ranking them, which approaches a *uniform* selection strategy. Theorem 2 formalizes this *sign collapse* by comparing the exact score with the linear surrogate score $U_z^{\texttt{Lion},\mathrm{lin}}$.

Theorem 2 (Sign collapse). *Condition on the frozen optimizer state and scoring reference $\hat{\mathbf{g}}_{\mathcal{V}}$, and let $\mathbf{g}_z \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$.*
(i) *(*Spread neutrality of the linear surrogate score.*)*
For a sign pattern $\mathbf{s}_t \triangleq \mathrm{sign}(\mathbf{m}_{t-1}) \in \{-1, +1\}^d$,

$$U_z^{\texttt{Lion},\mathrm{lin}} \triangleq \eta_t \langle \mathbf{g}_z, \mathbf{s}_t \odot \hat{\mathbf{g}}_{\mathcal{V}} \rangle \quad \Longrightarrow \quad \mathrm{Var}\big[U_z^{\texttt{Lion},\mathrm{lin}}\big] = \eta_t^2 \sum_{j=1}^{d} \sigma_j^2 [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2 = \mathrm{Var}\big[U_z^{\texttt{SGD}}\big]. \quad (5)$$

(ii) *(*Exponential collapse of the exact score.*) Suppose $\exists \kappa \geq 1$ satisfying Assumption 1, the exact* `Lion` *score $U_z^{\texttt{Lion}} \triangleq \eta_t \langle \mathrm{sign}\big(\beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_z\big), \hat{\mathbf{g}}_{\mathcal{V}} \rangle$ satisfies*

$$\mathrm{Var}\big[U_z^{\texttt{Lion}}\big] \leq 8 e^{-\kappa^2/2} \eta_t^2 \, \|\hat{\mathbf{g}}_{\mathcal{V}}\|_2^2 . \quad (6)$$

Consequently, when $\left\|\boldsymbol{\Sigma}^{1/2} \hat{\mathbf{g}}_{\mathcal{V}}\right\|_2 > 0$ and $\kappa^2 \geq 2 \log\left(\frac{8\|\hat{\mathbf{g}}_{\mathcal{V}}\|_2^2}{\left\|\boldsymbol{\Sigma}^{1/2}\hat{\mathbf{g}}_{\mathcal{V}}\right\|_2^2}\right)$, we have

$$\mathrm{Var}\big[U_z^{\texttt{Lion}}\big] \leq \mathrm{Var}\big[U_z^{\texttt{Lion},\mathrm{lin}}\big] = \mathrm{Var}\big[U_z^{\texttt{SGD}}\big]. \quad (7)$$

Theorem 2 (i) shows that a frozen sign pattern preserves the `SGD` score variance but cannot adaptively amplify discriminability; Theorem 2 (ii) shows that the exact score can degrade this baseline spread under momentum dominance. The full proof are deferred to Appendix D.1.

Diagonal-adaptive advantage. Theorem 3 further compares the optimizers along their linear diagonal operators. We define the coordinate weights $w_j \triangleq \dfrac{\sigma_j^2 [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2}{\sum_k \sigma_k^2 [\hat{\mathbf{g}}_{\mathcal{V}}]_k^2}$ with variance ratio $\dfrac{\mathrm{Var}[U_z^{(\boldsymbol{\pi})}]}{\mathrm{Var}[U_z^{\texttt{SGD}}]} = \sum_{j=1}^{d} w_j \pi_j^2$ to compare the benefits of linear operators against `SGD`.

Theorem 3 (Diagonal-adaptive advantage, and the discriminability ranking). *Under Assumption 2 and its curvature analogue for* `Sophia` *with its hessian estimates,*

$$\frac{\mathrm{Var}[U_z^{\texttt{AdamW}}]}{\mathrm{Var}[U_z^{\texttt{SGD}}]} = \sum_{j=1}^{d} w_j \, [\mathbf{c}_t]_j^2 > 1 - \mathrm{O}(\varepsilon'), \qquad \frac{\mathrm{Var}[U_z^{\texttt{Sophia}}]}{\mathrm{Var}[U_z^{\texttt{SGD}}]} > 1 - \mathrm{O}(\varepsilon''), \quad (8)$$

with $\varepsilon', \varepsilon''$ the respective numerical floors. Combined with Theorem 2, in the regime $\kappa^2 \geq 2\log\big(8\,\|\mathbf{g}_{\mathcal{V}}\|_2^2 \,/\, \left\|\boldsymbol{\Sigma}^{1/2}\mathbf{g}_{\mathcal{V}}\right\|_2^2\big)$ the spreads of the operator classes are ranked:

$$\underbrace{\mathrm{Var}[U_z^{\texttt{Lion}}]}_{\textit{exact sign}} \leq \underbrace{\mathrm{Var}[U_z^{\texttt{Lion},\mathrm{lin}}]}_{\textit{linearized sign}} = \mathrm{Var}[U_z^{\texttt{SGD}}] < \mathrm{Var}[U_z^{\texttt{AdamW}}], \ \mathrm{Var}[U_z^{\texttt{Sophia}}].$$

The final inequality holding separately for each of the two diagonal-adaptive scores from `AdamW` and `Sophia`, which the *discriminability* does not order against each other.

Radial collapse in the ideal polar geometry of `Muon`. The ideal `Muon` map removes the stretching of the momentum matrix while retaining its orientation. To see this, write the polar decomposition as $\mathbf{A}_t = \mathbf{Q}\mathbf{H}$: the ideal update is $\mathbf{Q} = \mathrm{msign}(\mathbf{A}_t)$, whereas $\mathbf{H} \succ 0$ describes the stretching that the map discards. A candidate can therefore change the stretching without turning the update. We call such changes *radial*; changing singular values while keeping their directions fixed is a simple example.

For a frozen full-column-rank momentum $\mathbf{A}_t$ and scoring reference $\mathbf{R} = \hat{\mathbf{G}}_{\mathcal{V}}$, decompose each candidate gradient into its radial component and the orthogonal remainder, $\mathbf{G}_z = \mathbf{G}_z^{\mathrm{rad}} + \mathbf{G}_z^{\tan}$. The latter is its *tangential* component, which can change the update's orientation. The precise decomposition is given in Appendix D.3. Let $\mathcal{L}_t = \mathrm{D}\,\mathrm{msign}(\mathbf{A}_t)$ denote the local linear response of the ideal map.

Theorem 4 (Radial selectivity of the ideal polar score). *At the frozen state, suppose $\mathbf{G}_z$ has finite second moment and pairwise uncorrelated entries in the singular-vector coordinates specified in Appendix D.3. For $\eta_t > 0$ and candidate weight $\alpha \geq 0$, define $U_z^{Muon} = \eta_t \alpha \langle \mathcal{L}_t[\mathbf{G}_z], \mathbf{R} \rangle_F$, $U_z^{\tan} = \eta_t \langle \mathbf{G}_z^{\tan}, \mathbf{R} \rangle_F$. Then U_z^{Muon} depends on the candidate only through $\mathbf{G}_z^{\tan}$, and*

$$\mathrm{Var}(U_z^{Muon}) \leq \frac{\alpha^2}{\sigma_{\min}(\mathbf{A}_t)^2} \mathrm{Var}(U_z^{\tan}), \tag{9}$$

where $\sigma_{\min}(\mathbf{A}_t) > 0$ is the smallest singular value of the frozen momentum.

Thus, two candidates that differ *only radially* receive the same first-order score. The collapse is selective: radial differences are ignored, while tangential differences are reweighted according to the momentum spectrum. The *practical* right-preconditioned proxy used in our experiments (2), behaves differently: it rescales the gradient without removing its radial component, and can therefore distinguish candidates that the ideal first-order score treats as identical. Theorem 4 implies neither an unconditional variance ordering against `SGD` nor a radial-collapse guarantee for the experimental proxy. The exact variance formulas, proof, and proxy discrepancy are given in Appendices D.3 and D.4.

5 Experiments

We conduct extensive experiments on LLM pretraining and give the first optimizer benchmarking along the *OptiSelect* paradigm (§5.1). Notably, we also show that the optimizer-aligned geometry may not be optimal for data selection (§5.2). We further assess the impact of data format on *OptiSelect* with rephrasing, showing that the benefits remains upon modern data synthesis pipeline (§5.3).

Models and training. We train 124M and 720M decoder-only transformers models following Llama architecture (Touvron et al., 2023) adopted in Semenov et al. (2025). We benchmark seven core optimizers, `AdamW`, `AdEMAMix`, `Sophia`, `Lion` and `D-Muon` under both standard training and *OptiSelect* paradigm, incurring a direct comparison of the effectiveness of online data selection. We train 124M and 720M models for 4.2B and 14B tokens respectively, matching their Chichilla (Hoffmann et al., 2022) optimal token amount. We apply the optimal learning rate tuned from a grid search on both 124M and 720M models, detailed in Appendix E.4. Complete configurations are provided in Appendix E.

Table 1: **Data selection benefits across optimizers.** Optimizers are measured according to their downstream losses and loss gap $\Delta = \mathcal{L}_{\text{standard}} - \mathcal{L}_{\text{OptiSelect}}$, followed by their ranks. Larger Δ indicates larger data selection gain.

	124M			720M		
Optimizer	standard	*OptiSelect*	Δ	standard	*OptiSelect*	Δ
`AdEMAMix`	3.410 (2)	3.274 (2)	**+0.135**	3.004 (3)	2.908 (2)	**+0.096**
`AdamW`	3.361 (1)	3.241 (1)	**+0.119**	2.995 (2)	2.891 (1)	**+0.104**
`Sophia`	3.486 (5)	3.458 (4)	**+0.028**	3.049 (5)	3.026 (5)	**+0.023**
`Lion`	3.417 (3)	3.399 (3)	**+0.018**	3.018 (4)	3.022 (4)	**−0.004**
`D-Muon`	3.455 (4)	3.471 (5)	**−0.015**	2.993 (1)	2.944 (3)	**+0.049**

Data and selection proxy. All the models are trained on FineWeb-HQ[2], a high-quality, model-filtered pretraining dataset derived as a subset of FineWeb. The selection proxy $\mathcal{V}$ is a fixed set of 4,096 densely packed sequences drawn from a *downstream* task mixture—ARC-Easy, ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA and SciQ, which serves as a targeted guidance of data selection. The proxy gradient is re-computed every 1,000 steps. The ablation of proxy-refresh interval is presented in Appendix E.5.

We report two axes of evaluation: (1) ***Downstream loss*** computes the loss of the correct continuation given the prompt on held-out validation sets, averaged across five proxy tasks; (2) ***Zero-shot accuracy*** as the exact match rate measured by `lm-evaluation-harness` (Gao et al., 2024) toolkit. We report the accuracy scores over nine tasks, including the 5 *in-domain* proxy tasks plus 4 *out-of-domain* ones WinoGrande, WSC, CommonsenseQA and OpenBookQA, which the proxy never sees.

5.1 The *OptiSelect* Benchmark: Selection Gain Across Optimizers and Scale

Optimizer ranking under *OptiSelect*. Figure 2 and Table 1 report the downstream losses across optimizers and model scales. The optimizer ranking presented in Table 1 demonstrates a scale-dependent impact from data selection: at 124M, the optimizer ranks on standard training are mostly

[2] https://huggingface.co/datasets/epfml/FineWeb-HQ

preserved on *OptiSelect*; while the ranking differs significantly at 720M. In terms of the downstream accuracy (Figure 3), despite the limited gain from online data selection, `D-Muon` consistently achieves the best averaged downstream accuracy, benefits from its extraordinary efficacy in optimization.

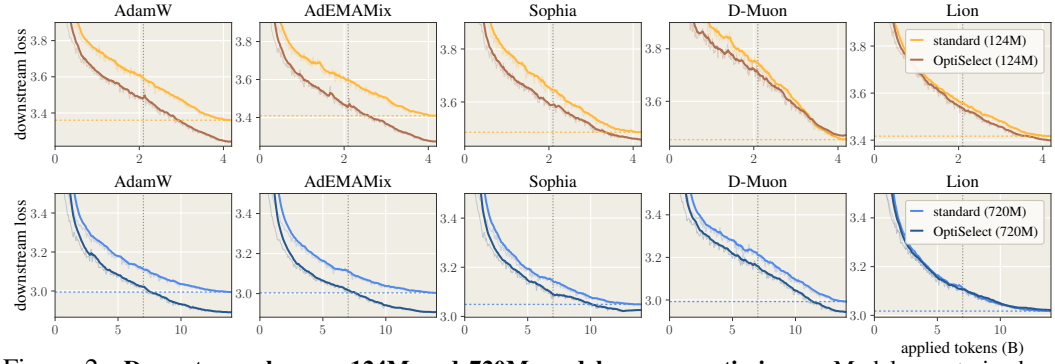


Figure 2: **Downstream loss on 124M and 720M models across optimizers.** Models are trained on 4.2B/14.03B tokens under standard and *OptiSelect* paradigm, respectively. The intersection of two dotted curves denotes the processed-matched baseline with equal processed tokens under $\lambda = 2$.

Empirical–theory consistency. According to Figure 2 and Table 1, the empirical selection gains remarkably *align with our theoretical analysis*: *OptiSelect* consistently improves the downstream loss with `AdamW`, `AdEMAMix` and `Sophia` at both 124M and 720M, whereas obtain no benefits with `Lion`. `D-Muon` demonstrates a scale-dependent behavior: at 124M, the selection gain shrinks by the end of training; while at 720M, the benefit remains significant and persists throughout the training.

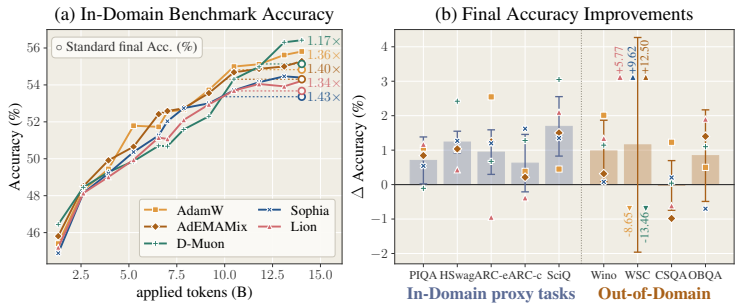


Figure 3: **Downstream Accuracy at 720M scale.** *OptiSelect* outperforms standard training across all optimizers on in-domain and OoD benchmarks.

***OptiSelect* gains in sample efficiency, not compute.** Figure 2 and Figure 3 expose a critical reflection: *OptiSelect*'s gains are measured upon optimizer *steps* at equal *applied* tokens while not the gains per token *read*. At $\lambda = 2$, every selection step reads twice the tokens it applies, each with a full forward–backward pass, therefore a fair comparison with equal read-tokens requires *OptiSelect* achieveing a $2\times$ acceleration in optimization steps. However, comparing with standard training with matched read tokens, *OptiSelect* falls behind in both downstream loss and accuracy across all five optimizers. Given the deficit shrinks at larger scale, *OptiSelect* is still promising on scaled-up experiments.

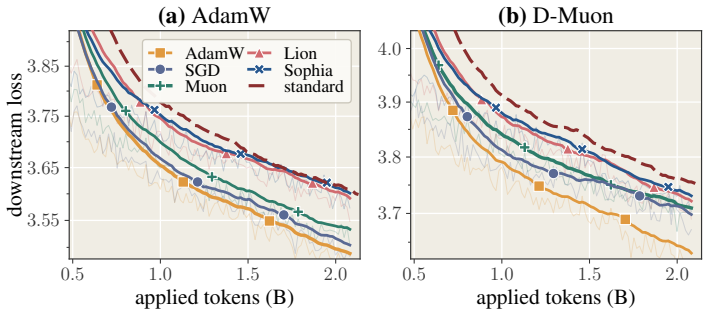


Figure 4: **Scoring geometry ablations under `AdamW` and `D-Muon` optimizers**. The matched geometry works best under `AdamW`, while both `AdamW` and `SGD` alternatives beat the matched scoring geometry on `D-Muon`.

5.2 Which Scoring Geometry Should an Optimizer Use?

The inconsistency of optimizer ranking between standard training and *OptiSelect* indicates a dichotomy of the preconditioning geometry for optimization and data selection: The geometry from the best-performed optimizer, `D-Muon`, is *not the optimal for data selection*. Inspired from the observation, we decouple the choice of optimizer and the geometry adopted for attribution scoring in (1). We run the scoring geometry ablation under two base optimizers `AdamW` and `D-Muon` (Figure 4).

Each optimizer is combined with various $\mathcal{O}_t$, i.e. the geometry, from each of the five optimizers. The `SGD` scoring geometry recovers the GREATS algorithm from Wang et al. (2024).

The optimizer-matched scoring geometry is not necessarily the best. Under `AdamW`, the best scoring geometry is also from `AdamW` which matches the optimization dynamics. However, with `D-Muon` as base optimizer, the pairing geometry does not work well in data selection, while the geometry of `AdamW` outperforms the others by a significant margin. Notably, **the second-momentum *diagonal-adaptive operator from* `AdamW` *serves as the best scoring geometry*** in both `AdamW` and `D-Muon` sweeps; and *scoring without preconditioner* (`SGD`) beats most of the scoring-geometry. Additionally, `AdamW` with the curvature-diagonal and sign-based geometries from `Sophia` and `Lion` can hardly outperform the standard training baseline with no selection. It indicates that a wrong geometry can hurt the performance of online data selection.

5.3 Does Selection Help with Rephrasing?

Selection and format rephrasing are two techniques on data quality, while to what extend they are collaborative or conflict is unexplored. FinePhrase (HuggingFaceFW, 2025) rewrites documents from FineWeb-edu (Lozhkov et al., 2024) into various *formats*, while leaving its content largely intact. We train `AdamW` at 124M on mixtures containing 0%, 50% and 100% rephrased documents with consistent setup as in §5.

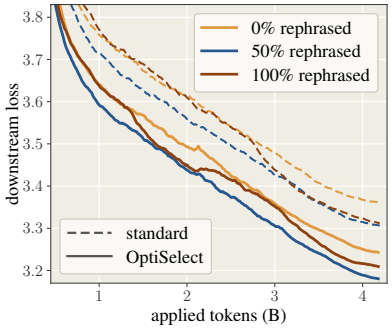


Figure 5: **Downstream loss curves across rephrased mixtures**, where the benefits from selection are consistent.

The selection benefit remains under data rephrasing. According to Figure 5, *OptiSelect* consistently improves the downstream loss by a large margin comparing to the standard training. *OptiSelect* also outperforms standard training across rephrasing mixtures on the average downstream accuracy (Figure 6, Table 10). It indicates that rephrasing and selection are not substitutes or conflict against each other.

***Format mismatching* is a confound for downstream accuracy, but it depends on evaluation harness pattern.** While the selection gets similar amount of gains on downstream loss along various rephrasing mixtures, the benefits on downstream accuracies move significantly along the *formatting* axis. Specifically, we separate the downstream tasks into QA- and completion-formatted categories according to the prompting format adopted in the evaluation harness (Appendix G.1). Under standard pretraining, the mixture with 50% rephrased documents achieves the best downstream loss and accuracy scores in both QA- and completion-formatted tasks, aligned with the common practice in (Maini et al., 2024) which samples real and synthetic data in a 1:1 ratio. In comparison, *OptiSelect* greatly augments the gain on QA-formatted accuracies, reaches 52.51% pp. On completion-formatted tasks, *OptiSelect* obtains a great improvement ($+2\%$ pp.) on the original FineWeb corpus, while increasing rephrasing proportions does not bring more benefits.

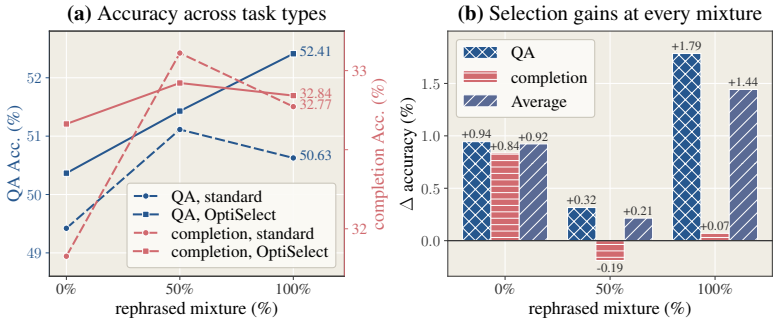


Figure 6: **Task accuracy on format rephrasing × selection at 124M. (a)** Accuracy against the rephrased mixture rates. **(b)** The selection gain at each mixture.

5.4 More Candidates, Better Performance?

The candidate multiplier λ is a critical hyperparameter for data selection, which determines the proportion of selected candidates. Instead of introducing as a magical hyperparameter settled in prior works (Wang et al., 2024; 2026), we quantify the effect of λ under the *selection gain principle* (Theorem 1). While the per-step gain $c_q\,\sigma_U$ with $q = 1/\lambda$ is monotonically increasing by λ, the overhead on the streamed and scored tokens grow correspondingly. Lemma 1 prices this trade-off and derives an optimal candidate multiplier $\lambda^\star$ in closed form.

Lemma 1 (Optimal candidate multiplier)**.** *Under the conditions of Theorem 1 with Gaussian candidate utilities, the selection gain per* streamed *token is*

$$\frac{c_q(\lambda)}{\lambda} \;=\; \phi\big(\Phi^{-1}(1-1/\lambda)\big), \tag{10}$$

which attains its unique maximum at $\lambda^\star = 2$*, of value* $\phi(0) = 1/\sqrt{2\pi}$.

Lemma 1 proves that the selection gain grows as $\sqrt{2\ln\lambda}$, monotonically but not linearly. We provide the full proof in D.5. Therefore, the increment of selection gain cannot compensate for its extra costs when it past the peak value at $\lambda^\star = 2$. *A token-limited budget should stream exactly twice what it trains on*, which is the default used everywhere else in this paper. In Figure 7, we empirically measure the raw selection gain Δ as the downstream loss reduction above the standard training and the gain per streamed token as Δ/λ, which validates the theory. Across various values of λ, the empirical raw selection gain Δ agrees with the theoretically computed values (Figure 7b), which keeps rising according to λ. In comparison, the measured Δ/λ demonstrates a concave curve, peaks at $\lambda^\star = 2$ while decreasing after the optimum.

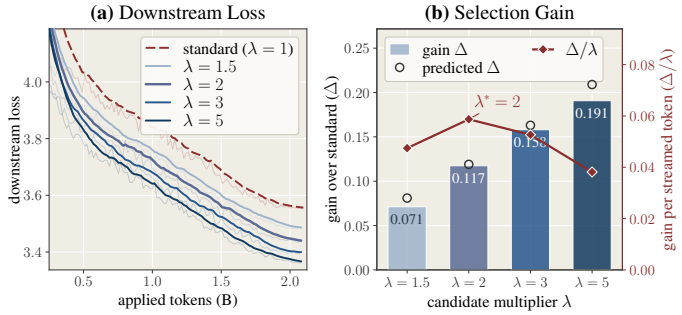


Figure 7: **The candidate-multiplier sweep at 124M. (a) Downstream loss curves. (b) Selection gain over the standard**. Open circles mark the theoretical values of $c_q\sigma_U$ from Theorem 1; the dark-red curve draws the gain per streamed token Δ/λ on the right axis, which peaks at $\lambda^\star = 2$.

6 Discussion and Limitations

The optimizer determines the landscapes of curriculum. Under *OptiSelect* the optimizer is not merely an optimization algorithm, it also serves as *the designer of the curriculum*. The variance normalization of `AdamW` builds a curriculum biased toward candidates whose gradients are large along rare dimensions, i.e. coordinates with small $[\hat{\mathbf{v}}_t]_j$ are amplified; `Muon`'s matrix-sign geometry builds a polar-tangent curriculum, which ignores candidates that only change singular values in the current polar basis and highlights those rotate singular subspaces. How to design an optimization algorithm that adapts to the data curriculum is a critical problem for future study.

What selection buys, and how it must be reported. Under a downstream proxy, *OptiSelect* consistently improves the metrics it targets as proxy while may not necessarily benefit for held-out tasks. According to Figure 3 (b), *OptiSelect* obtains benefits across 3 out of 4 Out-of-Domain benchmarks, while *OptiSelect* yields little gains on Commonsense QA task. Besides, selection is ahead at equal *applied* tokens and behind at equal tokens *read*, at 124M and at 720M alike: it converts extra forward–backward compute into fewer optimizer steps for the same quality, not into fewer tokens consumed. We report both matchings throughout and recommend that future evaluations of in-batch selection do the same; either number quoted alone misstates the method in one direction.

Limitations and future work. Our benchmark covers only 124M and 720M models, which can only predict the behaviour at multi-billion scale by extrapolation. Besides, we only conducted each run on a single random seed without claiming the variance across several seeds, mainly constrained from the high computation costs from pretraining. The scope of the proxy target is also limited to the general reasoning tasks, which are within the capability of small-scale models. We leave the exploration on more challenging domains, including Math, Code, and Agentic tasks for the future works.

7 Conclusion

In this work, we formalized *OptiSelect* as the training paradigm with optimizer-dependent online in-batch selection. We provide both theoretical analysis and empirical benchmarking on how various optimization geometries can affect the preference in data selection. Our findings on *OptiSelect* provide theoretical foundations and empirical insights for the next generation of pretraining optimizers, co-designed with the modern data deployment.

AI DISCLOSURE

In this work, we **USED** generative AI tools for paper polishing and proof writing assistance, analyzing existing literature and coding assistance. We have **NOT** used generative AI tools for formulating research or proof ideas, developing theoretical models or conceptual frameworks and synthetic data generation, synthetic evaluation are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendix B specifies the optimizer update rules and scoring operators, including implementation approximations. Appendices C–D provide the theoretical assumptions and proofs. Appendix E documents model architectures, training configurations, selection proxies, and hyperparameter sweeps. Appendices F–G report detailed experimental results. We will open-source our code at https://anonymous.4open.science/r/OptiSelect-ICLR27-640B/ for reproducibility.

ETHICS STATEMENT

This work studies the interleaving dynamics between optimizers and data selection using public available datasets and evaluation protocols. We identify no additional ethical concerns specific to this study.

REFERENCES

Rohan Anil, Vineet Gupta, Tomer Koren, Kevin Regan, and Yoram Singer. Scalable second order optimization for deep learning. arXiv preprint arXiv:2002.09018, 2020.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference On Machine Learning*, pp. 41–48, 2009.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. arXiv preprint arXiv:2409.20325, 2024.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *International Conference On Machine Learning*, pp. 560–569, 2018.

Tyler A. Chang, Dheeraj Rajagopal, Tolga Bolukbasi, Lucas Dixon, and Ian Tenney. Scalable influence and fact tracing for large language model pretraining. In *International Conference On Learning Representations*, 2025.

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*, pp. 49205–49233, 2023.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, Jeff Schneider, Eduard Hovy, Roger Grosse, and Eric Xing. What is your data worth to GPT? LLM-scale data valuation with influence functions. In *Advances In Neural Information Processing Systems*, pp. 162015–162056, 2025.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. arXiv preprint arXiv:1803.05457, 2018.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. DoGE: Domain reweighting with generalization estimation. In *International Conference On Machine Learning*, pp. 12895–12915, 2024.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness. Zenodo, 2024. Version v0.4.3.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference On Machine Learning*, pp. 1842–1850, 2018.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Advances In Neural Information Processing Systems*, pp. 30016–30030, 2022.

HuggingFaceFW. FinePhrase. Hugging Face dataset, 2025.

Keller Jordan. Muon: An optimizer for hidden layers in neural networks. Blog post, 2024.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International Conference On Machine Learning*, pp. 2525–2534, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference On Learning Representations*, 2015.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference On Machine Learning*, pp. 1885–1894, 2017.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. DataComp-LM: In search of the next generation of training sets for language models. In *Advances In Neural Information Processing Systems*, pp. 14200–14282, 2024.

Jingru Li, Yibo Fan, and Huan Li. Variance-adaptive Muon: Accelerating LLM pretraining with NSR-modulated and variance-scaled momentum. arXiv preprint arXiv:2601.14603, 2026.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *International Conference On Learning Representations*, 2024.

Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. arXiv preprint arXiv:1511.06343, 2015.

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

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. FineWeb-Edu: The finest collection of educational content. Hugging Face dataset, 2024.

Pratyush Maini, Skyler Seto, Richard Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *Annual Meeting Of The Association For Computational Linguistics*, pp. 14044–14072, 2024.

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

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Mądry. TRAK: Attributing model behavior at scale. In *International Conference On Machine Learning*, pp. 27074–27113, 2023.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances In Neural Information Processing Systems*, pp. 30811–30849, 2024.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances In Neural Information Processing Systems*, pp. 19920–19930, 2020.

Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. Benchmarking optimizers for large language model pretraining. arXiv preprint arXiv:2509.01440, 2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. SOAP: Improving and stabilizing Shampoo using Adam for language modeling. In *International Conference On Learning Representations*, 2025.

Jiachen T Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. GREATS: Online selection of high-quality data for LLM training in every iteration. In *Advances In Neural Information Processing Systems*, pp. 131197–131223, 2024.

Jiachen T Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. Data Shapley in one training run. In *International Conference On Learning Representations*, 2025.

Shaobo Wang, Xuan Ouyang, Tianyi Xu, Yuzheng Hu, Jialin Liu, Guo Chen, Tianyu Zhang, Junhao Zheng, Kexin Yang, Xingzhang Ren, Dayiheng Liu, and Linfeng Zhang. OPUS: Towards efficient and principled data selection in large language model pre-training in every iteration. In *International Conference On Machine Learning*, 2026.

Kaiyue Wen, David Leo Wright Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. In *International Conference On Learning Representations*, 2026.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *International Conference On Machine Learning*, pp. 54104–54132, 2024.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. In *Advances In Neural Information Processing Systems*, pp. 69798–69818, 2023a.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In *Advances In Neural Information Processing Systems*, pp. 34201–34227, 2023b.

Zichun Yu, Spandan Das, and Chenyan Xiong. MATES: Model-aware data selection for efficient pretraining with data influence models. In *Advances In Neural Information Processing Systems*, pp. 108735–108759, 2024.

Huizhuo Yuan, Yifeng Liu, Shuang Wu, Xun Zhou, and Quanquan Gu. MARS: Unleashing the power of variance reduction for training large models. In *International Conference On Machine Learning*, pp. 73553–73587, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Annual Meeting Of The Association For Computational Linguistics*, pp. 4791–4800, 2019.

Appendix

CONTENTS

1 Introduction **1**

2 Related Work **2**

3 The *OptiSelect* Paradigm **2**

3.1 Data Schemes in LLM Pretraining 2

3.2 Optimizer-Aware Influence Scoring 3

4 What Makes an Optimizer Good at Selecting Data? **4**

5 Experiments **6**

5.1 The *OptiSelect* Benchmark: Selection Gain Across Optimizers and Scale 6

5.2 Which Scoring Geometry Should an Optimizer Use? 7

5.3 Does Selection Help with Rephrasing? 8

5.4 More Candidates, Better Performance? 8

6 Discussion and Limitations **9**

7 Conclusion **9**

A Extended Related Work **16**

A.1 Online Data Selection and Curriculum Learning 16

A.2 Influence Functions and Data Attribution 16

A.3 Modern Optimizers for LLM Pretraining 17

B Benchmarked Optimizers: Update Rules and Frozen-State Operators **18**

B.1 `SGD` with Momentum 18

B.2 `AdamW` 18

B.3 `AdEMAMix` 19

B.4 `Sophia` 19

B.5 `Lion` 20

B.6 `Signum` 20

B.7 `D-Muon` 21

B.8 Classification of the Frozen-State Operators 22

C The Selection-Gain Principle: Proof and Empirical Evidence **23**

C.1 Proof of Theorem 1 23

D Detailed Proofs for *Discriminability* (§4) **27**

D.1 Sign Collapse (Theorem 2) 27

D.2 Diagonal-Adaptive Advantage of `AdamW` and `Sophia` (Theorem 3) 30

D.3 Radial Collapse of `Muon` (Theorem 4) 32

D.4 Right-Preconditioned `Muon` Proxy and Exact Error 35

D.5 Optimal Candidate Multiplier (Lemma 1) 36

E Experiment Configurations **37**

E.1 Model Architecture 37

E.2 Training Arguments 37

E.3 Selection Proxy Details 37

E.4 Learning Rate Sweep 37

E.5 Proxy Refresh Interval 39

F Extended Experimental Results **40**

F.1 Complete scores at 124M 40

F.2 Complete scores at 720M 42

G Data Selection under Rephrasing **43**

G.1 Details on the Downstream Tasks 43

G.2 Extended Results on Rephrasing 43

A Extended Related Work

This section gives a detailed description of related work mentioned in §2. The organizing claim is unchanged: existing work either selects data while treating the optimizer as given, or studies optimizers while treating the data stream as given, and the two literatures barely cite each other. OptiSelect sits at their intersection.

A.1 Online Data Selection and Curriculum Learning

The idea that training order and sample weighting matter predates deep learning. Bengio et al. (2009) proposed easy-to-hard curricula for neural network training; Loshchilov & Hutter (2015) ranked samples by their current loss and sampled with exponential decay, reporting a $5\times$ speedup on MNIST; Katharopoulos & Fleuret (2018) formalized the problem as importance sampling, training an auxiliary model to predict upper bounds on per-sample gradient norms. These methods score samples by scalar losses or raw gradient norms; none accounts for the transformation the optimizer applies to a gradient before it becomes an update, which is the object this paper studies.

For LLM-scale pretraining, data selection has largely been *static*: the corpus is curated once, before training, by a criterion that never sees the model state. DSIR (Xie et al., 2023b) performs importance resampling in a hashed n-gram feature space to match a target distribution; DCLM (Li et al., 2024) trains fastText classifiers for quality filtering and shows that classifier choice dominates other curation decisions at scale; FineWeb-Edu (Penedo et al., 2024) distills LLM quality judgments into a lightweight scorer (the FineWeb family also supplies FineWeb-HQ, the corpus our benchmark streams); DoReMi (Xie et al., 2023a) optimizes domain mixture weights with a small proxy model and group-DRO objective. Static filters are cheap at training time and composable with any optimizer, but they cannot adapt to what the model currently knows, and they answer a different question from ours: *what* to train on, rather than *what next*. A complementary static axis is rewriting rather than filtering: FinePhrase (HuggingFaceFW, 2025) rephrases FineWeb documents into cleaner prose, changing the format of the text while preserving content. Section 5.3 shows this axis composes with online selection rather than substituting for it: the selection gain is essentially unchanged at 0%, 50% and 100% rephrased shares.

Online selection interleaves scoring with training. MATES (Yu et al., 2024) is the closest precursor to our setting at pretraining scale: it performs influence-guided selection during pretraining, but predicts influence with a small auxiliary model refreshed periodically, so the scores reflect the optimizer only indirectly through the checkpoints the auxiliary model is fitted to. GREATS (Wang et al., 2024) selects greedily within each batch using exact per-sample gradient inner products made affordable by the Ghost trick, at fine-tuning scale. OPUS (Wang et al., 2026) is the most principled online approach and the direct foundation of our benchmark: it computes optimizer-aware influence scores from per-sample gradients at every step and reports up to $2\times$ data efficiency at $\lambda = 2$. OPUS, however, instantiates and validates the paradigm for `AdamW` and `Muon` only, and offers little analysis of *how* the operator shapes selection; the paradigm's defining component—the optimizer—is exactly the axis it leaves unexplored, and the axis along which this paper varies everything.

A.2 Influence Functions and Data Attribution

Classical influence functions (Koh & Liang, 2017) estimate the effect of removing training point z on the loss at a test point z' as $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(z')^\top H^{-1}\nabla_{\boldsymbol{\theta}}\mathcal{L}(z)$, requiring an inverse-Hessian-vector product that is intractable at LLM scale and fragile in deep non-convex models. A line of work replaces the Hessian with tractable surrogates. TracIn (Pruthi et al., 2020) approximates influence by gradient dot products summed over checkpoints, which is exact for `SGD` dynamics and explicitly ignores optimizer preconditioners. TRAK (Park et al., 2023) scales attribution to billion-parameter models by random projection and kernel linearization. Data Shapley valuations can now be computed in a single training run (Wang et al., 2025), and LoGra (Choe et al., 2025) makes influence queries affordable at GPT scale by projecting gradients onto low-rank subspaces during training.

Most relevant to us is the recent recognition that the optimizer's adaptive state fundamentally changes attribution. LESS (Xia et al., 2024) extends TracIn to account for `Adam`'s second moment when selecting instruction-tuning data, finding that `Adam`-aware selection consistently outperforms `SGD`-aware selection under an `Adam`-trained model. TrackStar (Chang et al., 2025) divides per-

sample gradients by Adam's $\sqrt{\hat{\mathbf{v}}_t}$ before scoring, reporting that uncorrected scores preferentially retrieve long, high-magnitude examples rather than genuinely influential ones. These works treat a single optimizer's correction as a fix to an attribution pipeline. Our frozen-state operator $\mathcal{O}_t$ (§3.2) generalizes that move into a formalism: every optimizer induces its own influence geometry, the induced geometries differ qualitatively (diagonal, sign, polar-tangent), and the differences are large enough to reorder which optimizers benefit from selection at all (§5). Attribution also differs from our setting in *when* it acts: it explains a trained model's behaviour retrospectively, whereas OptiSelect scores candidates prospectively, before the update they would cause, with the optimizer state frozen at the current step.

A.3 Modern Optimizers for LLM Pretraining

The families. Adam (Kingma & Ba, 2015) with decoupled weight decay (Loshchilov & Hutter, 2019) remains the default for LLM pretraining. AdEMAMix (Pagliardini et al., 2025) adds a third, much slower momentum buffer ($\beta_3 \approx 0.9999$, mixed in with weight α), on the observation that gradients stay informative for tens of thousands of steps. Sophia (Liu et al., 2024) maintains a diagonal Hessian estimate via Hutchinson or Gauss–Newton–Bartlett probes and clips the preconditioned update coordinate-wise. The sign family descends from SignSGD and its momentum variant Signum (Bernstein et al., 2018), motivated by communication compression; Lion (Chen et al., 2023), discovered by program search, interpolates momentum and gradient inside the sign and stores the buffer with a separate decay. Muon (Jordan, 2024) applies Newton–Schulz orthogonalization to the momentum matrix of each 2-D parameter, approximating the matrix-sign/polar-factor update; the D-Muon variant we benchmark follows Semenov et al. (2025) in handling 1-D parameters with AdamW, and refinements continue to appear (Li et al., 2026). The Kronecker-factored line runs from Shampoo (Gupta et al., 2018) through its scalable variants (Anil et al., 2020) to SOAP (Vyas et al., 2025), which runs Adam in Shampoo's eigenbasis. Closest in spirit to our operator view, Bernstein & Newhouse (2024) read modern optimizers as steepest descent under different norms; our frozen-state operator is the same structural move aimed at a different question—not what update geometry an optimizer imposes, but what *influence* geometry it induces on data.

Benchmarks. Semenov et al. (2025) benchmark eleven optimizers (AdamW, AdEMAMix, MARS, D-Muon, SOAP, ADOPT, Prodigy, Sophia, Lion, Signum, SF-AdamW) on Llama-style models up to 720M parameters on FineWeb—the recipe our setup follows for comparability—finding that AdEMAMix dominates at larger scale, sign-based methods degrade with model size, SOAP wins short horizons but not long ones, and Sophia diverges in small-batch regimes. The concurrent study of Wen et al. (2026) partially disputes Muon's ranking and finds per-component learning-rate tuning critical. Both benchmarks train on uniformly sampled batches; neither considers any form of data selection. Our results give a concrete reason this omission matters: the benefit an optimizer draws from selection is an ordering of its own, stable across scale and unpredicted by standard-training quality (§5)—an optimizer chosen from these leaderboards may be the wrong choice for a selection pipeline.

B BENCHMARKED OPTIMIZERS: UPDATE RULES AND FROZEN-STATE OPERATORS

This section specifies the optimizer update rules and the frozen-state operators used for data selection. For each optimizer discussed below, we provide (i) pseudocode for the update rule, (ii) the corresponding frozen-state scoring operator $\mathcal{O}_t$, and (iii) the approximations and normalization conventions used for scoring. We distinguish the implemented scoring maps from the exact nonlinear maps and derivative-based operators considered in the analysis.

Derivation conventions. We use the following conventions:

(S1) *Freeze the scoring state.* Hold the reference state $\mathcal{S}_t$ fixed across candidates at step t, and write $\mathbf{x}$ for the gradient argument. Where scoring also freezes a quantity that the optimizer normally updates using the incoming gradient, we state that approximation explicitly.

(S2) *Separate candidate-independent additive terms.* When the frozen update is affine in $\mathbf{x}$, separate its constant component from the candidate-dependent direction. An additive constant in a completed scalar score does not affect rankings or Boltzmann probabilities. Terms inside nonlinear maps cannot generally be removed this way; the scoring conventions for those maps are specified below.

(S3) *Choose a linear scoring map.* A nonlinear batch update generally cannot be decomposed into independent per-sample directions. We therefore distinguish derivative-based linearizations from linear surrogates: the analysis considers the ideal polar derivative for `Muon`, while the experiments use a right-preconditioned `Muon` proxy and fixed-sign surrogates for `Lion`/`Signum`. Exact additivity of these scoring maps does not imply exact attribution under the full optimizer update.

Multiplying a completed score by a common positive scalar preserves hard rankings but rescales its variance and changes raw-score Boltzmann sampling at fixed temperature. Rescaling $\mathcal{O}_t$ need not preserve the rankings induced by (1): when $\mathbf{G}_t$ is rescaled consistently, the alignment term scales linearly and the redundancy term quadratically. The omitted prefactors therefore define the scoring convention rather than an invariance of the full utility or optimizer update.

B.1 SGD WITH MOMENTUM

Algorithm 2 `SGD` with heavy-ball momentum, one step.

Require: learning rate η_t; momentum μ; first-order momentum buffer $\mathbf{b}$ (init $\mathbf{0}$)

$\mathbf{b}_t \leftarrow \mu\,\mathbf{b}_{t-1} + \hat{\mathbf{g}}_t$

$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta_t\,\mathbf{b}_t$

Derivation. Substituting $\mathbf{x}$ for $\hat{\mathbf{g}}_t$ (S1), the update direction is $\mathbf{d}(\mathbf{x}) = \mu\,\mathbf{b}_{t-1} + \mathbf{x}$. The frozen buffer term $\mu\,\mathbf{b}_{t-1}$ is candidate-independent and drops (S2), leaving the identity map (S3):

$$\mathcal{O}_t^{\mathrm{SGD}}(\mathbf{x}) \;=\; \mathbf{x}. \tag{11}$$

`SGD` scoring operator. $\mathcal{O}_t^{\mathrm{SGD}}(\mathbf{x}) = \mathbf{x}$: scoring with raw gradients. With the momentum state fixed, the update direction is affine in the gradient. After removing the constant momentum term, the candidate-dependent map is exactly linear.

B.2 ADAMW

Algorithm 3 `AdamW` (Kingma & Ba, 2015; Loshchilov & Hutter, 2019), one step.

Require: learning rate η_t; decays (β_1, β_2); ε; decoupled weight decay λ; state with first- and second-moment estimates $(\mathbf{m}, \mathbf{v})$ (init $\mathbf{0}$)

$\mathbf{m}_t \leftarrow \beta_1\,\mathbf{m}_{t-1} + (1-\beta_1)\,\hat{\mathbf{g}}_t$

$\mathbf{v}_t \leftarrow \beta_2\,\mathbf{v}_{t-1} + (1-\beta_2)\,\hat{\mathbf{g}}_t^{\odot 2}$

$\hat{\mathbf{m}}_t \leftarrow \mathbf{m}_t/(1-\beta_1^t)$; $\quad \hat{\mathbf{v}}_t \leftarrow \mathbf{v}_t/(1-\beta_2^t)$

$\boldsymbol{\theta}_{t+1} \leftarrow (1-\eta_t\lambda)\,\boldsymbol{\theta}_t - \eta_t\,\hat{\mathbf{m}}_t \oslash (\sqrt{\hat{\mathbf{v}}_t} + \varepsilon)$

Derivation. For scoring, freeze $\mathbf{m}_{t-1}$ and the reference second-moment estimate $\hat{\mathbf{v}}_t$ across candidates. This suppresses the dependence of the denominator on the candidate gradient in the full `AdamW` update. Under this approximation, substituting $\mathbf{x}$ gives

$$\mathbf{d}(\mathbf{x}) \;=\; \underbrace{\tfrac{1-\beta_1}{1-\beta_1^t}\,\mathbf{x} \oslash (\sqrt{\hat{\mathbf{v}}_t}+\varepsilon)}_{\text{candidate-dependent}} \;+\; \underbrace{\tfrac{\beta_1}{1-\beta_1^t}\,\mathbf{m}_{t-1} \oslash (\sqrt{\hat{\mathbf{v}}_t}+\varepsilon) \;+\; \lambda\boldsymbol{\theta}_t}_{\text{constant for all candidates; drops by S2}}.$$

After removing the frozen additive terms, we set the current-gradient coefficient $(1-\beta_1)/(1-\beta_1^t)$ to one as a scoring normalization, obtaining the diagonal map below.

`AdamW` scoring operator. The scoring operator is linear, with exact per-sample decomposition under the frozen-denominator, unit-prefactor convention. $\mathcal{O}_t^{\texttt{AdamW}}(\mathbf{x}) = \mathbf{c}_t \odot \mathbf{x}$ with $\mathbf{c}_t = \mathbf{1} \oslash (\sqrt{\hat{\mathbf{v}}_t}+\varepsilon)$: inverse-square-root second-moment rescaling, assigning larger weights to coordinates with smaller estimated second moments.

B.3 AdEMAMix

Algorithm 4 `AdEMAMix` (Pagliardini et al., 2025), one step. The runs use the linear α- and β_3-warmup schedulers of the original paper.

Require: learning rate η_t; decays $(\beta_1, \beta_2, \beta_3)$; slow-buffer weight α; ε; weight decay λ; state $(\mathbf{m}^{(1)}, \mathbf{m}^{(2)}, \mathbf{v})$

$\mathbf{m}_t^{(1)} \leftarrow \beta_1\,\mathbf{m}_{t-1}^{(1)} + (1-\beta_1)\,\hat{\mathbf{g}}_t$ {fast buffer, bias-corrected below}

$\mathbf{m}_t^{(2)} \leftarrow \beta_3\,\mathbf{m}_{t-1}^{(2)} + (1-\beta_3)\,\hat{\mathbf{g}}_t$ {slow buffer, uncorrected}

$\mathbf{v}_t \leftarrow \beta_2\,\mathbf{v}_{t-1} + (1-\beta_2)\,\hat{\mathbf{g}}_t^{\odot 2}$; $\hat{\mathbf{m}}_t^{(1)} \leftarrow \mathbf{m}_t^{(1)}/(1-\beta_1^t)$; $\hat{\mathbf{v}}_t \leftarrow \mathbf{v}_t/(1-\beta_2^t)$

$\boldsymbol{\theta}_{t+1} \leftarrow (1-\eta_t\lambda)\,\boldsymbol{\theta}_t - \eta_t\,\big(\hat{\mathbf{m}}_t^{(1)} + \alpha\,\mathbf{m}_t^{(2)}\big) \oslash (\sqrt{\hat{\mathbf{v}}_t}+\varepsilon)$

Derivation. Under the same frozen-denominator convention, the candidate enters both the fast and slow buffers, yielding the update direction:

$$\mathbf{d}(\mathbf{x}) \;=\; \Big(\tfrac{1-\beta_1}{1-\beta_1^t} + \alpha\,(1-\beta_3)\Big)\,\mathbf{x} \oslash (\sqrt{\hat{\mathbf{v}}_t}+\varepsilon) \;+\; \text{frozen terms},$$

After removing the frozen additive terms and setting the displayed current-gradient coefficient to one, we obtain the same diagonal scoring form as `AdamW`, evaluated at `AdEMAMix` 's frozen second moment.

`AdEMAMix` scoring operator. $\mathcal{O}_t^{\texttt{AdEMAMix}}(\mathbf{x}) = \mathbf{c}_t \odot \mathbf{x}$, with the same functional form as `AdamW` under the stated normalization. The two methods need not produce identical scores: their update dynamics can lead to different parameter trajectories and second-moment estimates.

B.4 Sophia

Algorithm 5 `Sophia-G` (Liu et al., 2024), one step. B_H is the batch size used by the Hessian estimator; the trust scale is $\gamma = \rho\,B_H$.

Require: learning rate η_t; decays (β_1, β_h); trust parameter ρ; floor ε; Hessian interval k_H; weight decay λ; state $(\mathbf{m}, \mathbf{h})$

if $t \equiv 0 \pmod{k_H}$ **then**

  draw $\hat{y} \sim p_{\boldsymbol{\theta}_t}(\cdot \mid x)$ for the mini-batch; $\mathbf{g}_s \leftarrow \nabla_{\boldsymbol{\theta}}\,\ell(x, \hat{y}; \boldsymbol{\theta}_t)$

  $\hat{\mathbf{h}} \leftarrow B_H\,\mathbf{g}_s \odot \mathbf{g}_s$ {Gauss–Newton–Bartlett estimate of $\mathrm{diag}(\mathbf{H})$}

  $\mathbf{h}_t \leftarrow \beta_h\,\mathbf{h}_{t-k_H} + (1-\beta_h)\,\hat{\mathbf{h}}$

end if

$\mathbf{m}_t \leftarrow \beta_1\,\mathbf{m}_{t-1} + (1-\beta_1)\,\hat{\mathbf{g}}_t$

$\boldsymbol{\theta}_{t+1} \leftarrow (1-\eta_t\lambda)\,\boldsymbol{\theta}_t - \eta_t\,\mathrm{clip}\big(\mathbf{m}_t \oslash \max\{\gamma\,\mathbf{h}_t,\,\varepsilon\},\,1\big)$ {coordinate-wise clip at magnitude 1}

`Sophia` estimates curvature from training gradients $\mathbf{g}_s$,

$$\hat{\mathbf{h}} \;=\; B_H\,\mathbf{g}_s \odot \mathbf{g}_s \quad \text{(GNB; the Hutchinson variant uses } \mathbf{u} \odot (\mathbf{H}\mathbf{u}),\ \mathbf{u} \sim \mathrm{Unif}\{\pm 1\}^d), \tag{12}$$

$$\mathbf{h}_t \;=\; \beta_h\,\mathbf{h}_{t-k_H} + (1-\beta_h)\,\hat{\mathbf{h}}.$$

Derivation. Freeze $\mathbf{h}_t$ and substitute $\mathbf{x}$ into the momentum (S1):

$$\mathbf{d}(\mathbf{x}) \;=\; \mathrm{clip}\Big(\big[\beta_1\,\mathbf{m}_{t-1} + (1-\beta_1)\,\mathbf{x}\big] \oslash \max\{\gamma\,\mathbf{h}_t,\,\varepsilon\},\;1\Big) \;+\; \lambda\,\boldsymbol{\theta}_t.$$

Implementation. Clipping makes the update nonlinear in $\mathbf{x}$. For scoring, the implementation omits clipping and uses the *unclipped* operator

$$\mathcal{O}_t^{\texttt{Sophia}}(\mathbf{x}) \;=\; \mathbf{x} \oslash \max\{\gamma\,\mathbf{h}_t,\,\varepsilon\}. \tag{13}$$

Equation (13) defines an unclipped scoring surrogate, with $\mathbf{h}_t$ frozen, the frozen momentum contribution removed, and the current-gradient coefficient $1-\beta_1$ set to one. Its per-sample decomposition is exact for this linear scoring map, not for the clipped optimizer update in general.

`Sophia` scoring operator. $\mathcal{O}_t^{\texttt{Sophia}}(\mathbf{x}) \;=\; \mathbf{x} \oslash \max\{\gamma\,\mathbf{h}_t,\,\varepsilon\}$. This linear surrogate assigns larger weights to coordinates with smaller estimated curvature, subject to the numerical floor.

B.5 LION

Algorithm 6 `Lion` (Chen et al., 2023), one step. Define two decay coefficients: β_1 mixes inside the sign, β_2 updates the stored buffer.

Require: learning rate η_t; decays (β_1, β_2); weight decay λ; state $\mathbf{m}$ (init $\mathbf{0}$)
1: $\mathbf{c}_t \leftarrow \beta_1\,\mathbf{m}_{t-1} + (1-\beta_1)\,\hat{\mathbf{g}}_t$ {one-off interpolation; *not* stored}
2: $\boldsymbol{\theta}_{t+1} \leftarrow (1-\eta_t\lambda)\,\boldsymbol{\theta}_t - \eta_t\,\mathrm{sign}(\mathbf{c}_t)$
3: $\mathbf{m}_t \leftarrow \beta_2\,\mathbf{m}_{t-1} + (1-\beta_2)\,\hat{\mathbf{g}}_t$

Derivation. Substituting $\mathbf{x}$ gives the exact frozen-state operator directly:

$$\mathcal{O}_t^{\texttt{Lion}}(\mathbf{x}) \;=\; \mathrm{sign}\big(\beta_1\,\mathbf{m}_{t-1} + (1-\beta_1)\,\mathbf{x}\big). \tag{14}$$

When the frozen momentum dominates the candidate contribution, the exact operator usually agrees with the candidate-independent pattern $\mathbf{s}_t = \mathrm{sign}(\mathbf{m}_{t-1})$. Candidate dependence then arises through coordinate sign flips. Under the Gaussian candidate model and momentum-dominance assumption of Theorem 2, these flips are rare tail events. The default $\beta_1 = 0.9$ alone does not establish this regime.

Implementation. The implementation instead uses a *fixed-sign linear surrogate*[3]: the frozen pattern $\mathbf{s}_t$ multiplies the candidate gradient coordinate-wise. The alignment component of the score is $U_{z,\mathrm{align}}^{\mathrm{Lion,lin}} = \eta_t\langle \mathrm{sign}(\mathbf{m}_{t-1}) \odot \mathbf{g}_z,\,\mathbf{g}_{\mathcal{V}}\rangle$. The full utility also includes the redundancy term in (1). The collapse bound for the exact sign score does not directly apply to this surrogate.

`Lion` scoring operator. Exact: $\mathcal{O}_t(\mathbf{x}) = \mathrm{sign}(\beta_1\mathbf{m}_{t-1} + (1-\beta_1)\mathbf{x})$. Implemented: the linear surrogate map $\mathbf{x} \mapsto \mathbf{s}_t \odot \mathbf{x}$, $\mathbf{s}_t = \mathrm{sign}(\mathbf{m}_{t-1})$ read from the stored buffer.

B.6 SIGNUM

Algorithm 7 `Signum` (Bernstein et al., 2018), one step.

Require: learning rate η_t; momentum decay μ; state $\mathbf{m}$ (init $\mathbf{0}$)
1: $\mathbf{m}_t \leftarrow \mu\,\mathbf{m}_{t-1} + (1-\mu)\,\hat{\mathbf{g}}_t$
2: $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta_t\,\mathrm{sign}(\mathbf{m}_t)$

Derivation. Substituting $\mathbf{x}$ for the incoming gradient gives

$$\mathcal{O}_t^{\texttt{Signum}}(\mathbf{x}) \;=\; \mathrm{sign}\big(\mu\,\mathbf{m}_{t-1} + (1-\mu)\,\mathbf{x}\big). \tag{15}$$

[3]The superscript lin denotes a linear surrogate, not a Taylor or Fréchet linearization of the sign map, whose derivative is zero almost everywhere.

This has the same form as the Lion operator when $\beta_1 = \beta_2 = \mu$, with the same coefficient controlling both the direction used for the update and the stored momentum. The gradient-driven updates coincide under this identification; any weight-decay conventions must also be matched. Unlike Lion, what is signed is the freshly *stored* buffer, so the frozen pattern's persistence across steps tracks μ rather than a separate memory coefficient.

Signum scoring operator. Exact: $\mathcal{O}_t(\mathbf{x}) = \operatorname{sign}(\mu\mathbf{m}_{t-1} + (1-\mu)\mathbf{x})$. Implemented: the fixed-sign linear surrogate $\mathbf{x} \mapsto \mathbf{s}_t \odot \mathbf{x}$, as for Lion, with $\mathbf{s}_t = \operatorname{sign}(\mathbf{m}_{t-1})$. For plain SignSGD ($\mu = 0$), the implementation uses SGD scoring as a fallback; this is a scoring convention, not the exact sign-update map.

B.7 D-Muon

Algorithm 8 D-Muon (Jordan, 2024; Semenov et al., 2025), one optimizer step. Settings used throughout this paper: $\mu = 0.95$, $k_{\mathrm{NS}} = 5$, $r = 0.2$, $\lambda = 0.1$, and nesterov = False.

Require: learning rate η_t; momentum μ; Newton–Schulz steps k_{NS}; RMS target r; weight decay λ; AdamW hyperparameters $(\beta_1, \beta_2, \varepsilon_a)$ for the auxiliary path

1: **Partition:** $\mathbf{W}$ takes the Muon path iff $\mathbf{W}$ is $\geq$ 2-D *and* $\dim_0(\mathbf{W}) < 10^4$; embeddings and the LM head (vocabulary-sized first axis) therefore take the AdamW path, as do all 1-D parameters.
2: **for** each Muon-path weight $\mathbf{W} \in \mathbb{R}^{m\times n}$ with gradient $\mathbf{G}_t$ **do**
3: $\quad \mathbf{M}_t \leftarrow \mu\,\mathbf{M}_{t-1} + \mathbf{G}_t$ {unnormalized heavy-ball buffer}
4: $\quad \mathbf{N}_t \leftarrow \mathbf{G}_t + \mu\,\mathbf{M}_t$ **if** nesterov **else** $\mathbf{M}_t$ {matrix entering the sign map}
5: $\quad \mathbf{O}_t \leftarrow \mathrm{NS}_{k_{\mathrm{NS}}}(\mathbf{N}_t) \approx \operatorname{msign}(\mathbf{N}_t) = \mathbf{U}\mathbf{V}^\top$ {quintic iteration, (16)}
6: $\quad \mathbf{W} \leftarrow (1-\eta_t\lambda)\,\mathbf{W} - \eta_t\,\sqrt{\max(m,n)}\;r\;\mathbf{O}_t$
7: **end for**
8: update the AdamW-path parameters by bias-corrected AdamW at the same η_t

D-Muon does not compute an SVD. It applies k_{NS} steps of a fixed odd polynomial iteration to approximately orthogonalize the update matrix:

$$\mathbf{X}_0 \;=\; \frac{\mathbf{N}_t}{\|\mathbf{N}_t\|_F + 10^{-7}}, \quad \mathbf{X}_{j+1} \;=\; a\,\mathbf{X}_j + \big(b\,\mathbf{X}_j\mathbf{X}_j^\top + c\,(\mathbf{X}_j\mathbf{X}_j^\top)^2\big)\mathbf{X}_j, \quad \mathrm{NS}_{k_{\mathrm{NS}}}(\mathbf{N}_t) \;\triangleq\; \mathbf{X}_{k_{\mathrm{NS}}}, \tag{16}$$

with $(a, b, c) = (3.4445,\ -4.7750,\ 2.0315)$. The finite iteration is distinct from the ideal polar map used in the analysis below.

Derivation. On a 2-D parameter block, the frozen state is the momentum matrix $\mathbf{A}_t = \mathbf{N}_t = \mathbf{M}_t$. For the ideal polar map, a candidate perturbation with weight α gives $\mathbf{G} \mapsto \operatorname{msign}(\mathbf{A}_t + \alpha\,\mathbf{G})$, which is nonlinear in $\mathbf{G}$. This is an idealized local perturbation model, not an identity for the finite Newton–Schulz update. The rigorous linear object is the Fréchet derivative of the polar map at $\mathbf{A}_t$,

$$\mathcal{O}_t^{\mathrm{Muon,lin}}(\mathbf{G}) \;=\; \mathrm{D}\operatorname{msign}(\mathbf{A}_t)[\mathbf{G}],$$

with full derivations and rank assumptions in Appendices D.3.1 and D.3. The derivative is defined per unit perturbation; a perturbation of size α contributes the corresponding scalar prefactor. Through the adjoint $\mathcal{T}_{\mathbf{A}_t}$ of (89), the first-order score is again a linear functional of the candidate, supported on singular-subspace rotations and new column-space directions and *exactly* zero on radial ones.

Implementation. The implementation scores with the cheaper *right-preconditioned proxy*

$$\mathcal{O}_t^{\mathrm{Muon,rp}}(\mathbf{G}) \;=\; \mathbf{G}\,(\mathbf{A}_t^\top\mathbf{A}_t + \varepsilon\mathbf{I})^{-1/2} \;=\; \mathbf{G}\,\mathbf{P}_t, \tag{17}$$

$\mathbf{P}_t$ is formed once per weight matrix by symmetric eigendecomposition and reused for R scoring calls ($R = 10$ in experiments). At a full-column-rank reference matrix and with $\varepsilon = 0$, this proxy agrees with $\mathrm{D}\operatorname{msign}(\mathbf{A}_t)$ on the column-space complement. Positive regularization changes that agreement. The proxy also retains radial candidate components that the ideal derivative annihilates; Appendix D.4 gives the discrepancy in closed form.

Table 2: Properties of the frozen-state maps discussed in this appendix. Linearity guarantees additivity of the represented batch direction, not exact attribution under the full optimizer update. "Magnitude information" refers to retaining continuous amplitude information rather than only sign patterns; it does not imply norm preservation. §The `Sophia` row describes its unclipped scoring surrogate. ‡The `Muon` row describes the ideal polar derivative; the implemented right-preconditioned proxy is also linear but retains radial components. *The `Lion`/`Signum` rows describe the exact sign maps; their implemented fixed-sign surrogates are linear and retain candidate amplitude on coordinates with nonzero frozen signs.

Optimizer	Operator family	Linear in x?	Magnitude information
Operators analyzed			
`SGD`	Unpreconditioned (identity)	✓	✓
`AdamW`	Diagonal adaptive (second moment)	✓	✓
`AdEMAMix`	Diagonal adaptive (second moment)	✓	✓
`Sophia`	Diagonal adaptive (curvature)	✓§	✓
`Muon`	Matrix sign / polar tangent	✓‡	Tangential; radial discarded
`Lion` *	Sign-based	×	×
`Signum` *	Sign-based	×	×

`D-Muon` scoring operator. The ideal polar derivative $\mathcal{O}_t^{\texttt{Muon},\mathrm{lin}}(\mathbf{G}) = \mathrm{D}\,\mathrm{msign}(\mathbf{A}_t)[\mathbf{G}]$ annihilates radial perturbations under the assumptions of Theorem 4. This radial-invariance result does not extend to the implemented proxy. Implemented: $\mathcal{O}_t^{\texttt{Muon},\mathrm{rp}}(\mathbf{G}) = \mathbf{G}(\mathbf{A}_t^\top \mathbf{A}_t + \varepsilon\mathbf{I})^{-1/2}$, with the preconditioner refreshed every $R = 10$ steps. The auxiliary `AdamW` path's parameters are excluded from scoring, as are all vocabulary-sized layers.

B.8 Classification of the Frozen-State Operators

Table 2 summarizes the operators discussed in this appendix, including the `SGD` and `Signum` reference methods. The following remark states what linearity of a frozen scoring map guarantees and distinguishes that property from fidelity to the optimizer update.

Remark 1 (Distributivity: when the per-sample decomposition is exact)**.** *The batch direction represented by a frozen linear scoring map decomposes exactly into per-sample directions:*

$$\mathcal{O}_t\left(\frac{1}{B}\sum_z \mathbf{g}_{z,t}\right) = \frac{1}{B}\sum_z \mathcal{O}_t(\mathbf{g}_{z,t}).$$

This holds for `SGD`, `AdamW`, `AdEMAMix`, and `Sophia`'s unclipped scoring operator (13). With a fixed scoring reference, these linear maps also induce linear scalar scores. By contrast, the exact clipped `Sophia` update, coordinate-wise sign maps, and the finite `Muon` matrix-sign map are nonlinear. We use the following linear scoring approximations: the fixed-sign surrogate for `Lion`/`Signum`, and the Fréchet derivative in the `Muon` analysis or the right-preconditioned proxy in the benchmark. Their per-sample decomposition is exact for the chosen scoring map, but only approximates attribution under the nonlinear optimizer update, to first order for $\mathcal{O}_t^{\texttt{Muon},\mathrm{lin}}$ and heuristically for the deployed sign and `Muon` proxies.

C THE SELECTION-GAIN PRINCIPLE: PROOF AND EMPIRICAL EVIDENCE

C.1 PROOF OF THEOREM 1

Let us separate two ingredients for the proof of Theorem 1. Smoothness of the validation objective relates the realized one-step loss reduction to the mean utility of the trained-on batch, while the Gaussian candidate model determines the expected utility of the uniformly sampled and top-selected batches.

Proof. Condition on the parameters $\boldsymbol{\theta}_t$, the optimizer state, and the frozen scoring reference at step t. Let

$$N \triangleq |\tilde{\mathcal{B}}_t| = \lambda B, \qquad q \triangleq \frac{B}{N} = \frac{1}{\lambda}.$$

For $a \in \{\mathrm{rand}, \mathrm{sel}\}$, denote by $\mathcal{B}_t^a \subseteq \tilde{\mathcal{B}}_t$ the batch returned by the corresponding sampling rule and by

$$\mathbf{d}_t^a \triangleq \mathcal{O}_t\left(\frac{1}{B}\sum_{z\in\mathcal{B}_t^a}\mathbf{g}_{z,t}\right) \tag{18}$$

the resulting batch direction.

We first make explicit the regularity conditions required for the one-step comparison. Assume that ∇F is $L_{F,t}$-Lipschitz on a neighborhood containing every line segment

$$\{\boldsymbol{\theta}_t - s\eta_t\mathbf{d}_t^a : 0 \le s \le 1\}, \qquad a \in \{\mathrm{rand}, \mathrm{sel}\},$$

and that the two batch directions satisfy

$$\max_{a\in\{\mathrm{rand},\mathrm{sel}\}} \mathbb{E}\left[\|\mathbf{d}_t^a\|_2^2\right] \le D_t^2 < \infty. \tag{19}$$

Assume further that the frozen map $\mathcal{O}_t$ used in the present argument is linear and is the same map used both to score candidates and to form the batch direction. Consequently,

$$\mathbf{d}_t^a = \frac{1}{B}\sum_{z\in\mathcal{B}_t^a}\mathcal{O}_t(\mathbf{g}_{z,t}). \tag{20}$$

The descent lemma now gives, for either sampling rule,

$$F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}_t - \eta_t\mathbf{d}_t^a) = \eta_t\langle \mathbf{g}_{\mathcal{V},t},\, \mathbf{d}_t^a\rangle + \rho_t^a, \tag{21}$$

where

$$|\rho_t^a| \le \frac{L_{F,t}\eta_t^2}{2}\|\mathbf{d}_t^a\|_2^2. \tag{22}$$

Let us clarify that this argument relies solely on local smoothness, requiring neither positive-definiteness nor a local quadratic-model assumption.

To connect (21) to the candidate utilities, write the frozen reference used in (1) as

$$\hat{\mathbf{g}}_{\mathcal{V},t} \triangleq \mathbf{g}_{\mathcal{V},t} - \eta_t\boldsymbol{r}_t, \tag{23}$$

where $\boldsymbol{r}_t$ is the frozen redundancy direction. For

$$\mathbf{u}_{z,t} \triangleq \mathcal{O}_t(\mathbf{g}_{z,t}),$$

the adopted utility score can then be written as

$$U(z,t) = \eta_t\langle \mathbf{u}_{z,t},\, \hat{\mathbf{g}}_{\mathcal{V},t}\rangle. \tag{24}$$

Averaging (24) over $\mathcal{B}_t^a$ and using (20) gives

$$\begin{aligned}\bar{U}_{\mathcal{B}_t^a} &\triangleq \frac{1}{B}\sum_{z\in\mathcal{B}_t^a}U(z,t)\\ &= \eta_t\langle \mathbf{d}_t^a,\, \mathbf{g}_{\mathcal{V},t}\rangle - \eta_t^2\langle \mathbf{d}_t^a,\, \boldsymbol{r}_t\rangle.\end{aligned} \tag{25}$$

Combining (21) and (25) therefore yields

$$F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}_t - \eta_t \mathbf{d}_t^a) = \bar{U}_{\mathcal{B}_t^a} + \varepsilon_t^a, \tag{26}$$

where

$$\varepsilon_t^a = \rho_t^a + \eta_t^2 \langle \mathbf{d}_t^a, \boldsymbol{r}_t \rangle. \tag{27}$$

By (19) and Cauchy–Schwarz,

$$|\mathbb{E}[\varepsilon_t^a]| \;\le\; \eta_t^2 C_t, \qquad C_t \;\triangleq\; \frac{L_{F,t}}{2} D_t^2 + \|\boldsymbol{r}_t\|_2 \, D_t. \tag{28}$$

Thus, uniformly over the two sampling rules,

$$R_t^a \;\triangleq\; \mathbb{E}[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}_t - \eta_t \mathbf{d}_t^a)] = \mathbb{E}\big[\bar{U}_{\mathcal{B}_t^a}\big] + \varepsilon_{t,a}, \qquad |\varepsilon_{t,a}| \le \eta_t^2 C_t. \tag{29}$$

Hence, up to a controlled second-order error, comparing one-step loss reductions is equivalent to comparing the expected mean utilities of the retained batches.

It remains to evaluate these two expectations. Let $U_1, \dots, U_N \overset{\text{i.i.d.}}{\sim} \mathcal{N}\big(\mu_U(t), \sigma_U^2(t)\big)$ be the candidate utilities, and let $U_{(1)} \le \cdots \le U_{(N)}$ denote their order statistics. Under uniform sampling, conditioning on the candidate pool gives

$$\mathbb{E}\Big[\bar{U}_{\mathcal{B}_t^{\text{rand}}} \,\Big|\, U_1, \dots, U_N\Big] = \frac{1}{N} \sum_{i=1}^{N} U_i. \tag{30}$$

Taking expectation once more yields the exact identity

$$\mathbb{E}\Big[\bar{U}_{\mathcal{B}_t^{\text{rand}}}\Big] = \mu_U(t). \tag{31}$$

For top-B selection,

$$\bar{U}_{\mathcal{B}_t^{\text{sel}}} = \frac{1}{B} \sum_{k=N-B+1}^{N} U_{(k)}. \tag{32}$$

Assume first that $\sigma_U(t) > 0$ and write

$$U_i = \mu_U(t) + \sigma_U(t) Z_i, \qquad Z_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

Then

$$\mathbb{E}\Big[\bar{U}_{\mathcal{B}_t^{\text{sel}}}\Big] = \mu_U(t) + c_{q,N} \sigma_U(t), \tag{33}$$

where the exact finite-pool coefficient is

$$c_{q,N} \;\triangleq\; \mathbb{E}\left[\frac{1}{B} \sum_{k=N-B+1}^{N} Z_{(k)}\right]. \tag{34}$$

The case $\sigma_U(t) = 0$ is immediate: all utilities are identical and both sampling rules have zero selection gain.

Equations (29), (31), and (33) give the exact finite-pool relations

$$\begin{aligned} R_t^{\text{rand}} &= \mu_U(t) + \varepsilon_{t,\text{rand}}, \\ R_t^{\text{sel}} &= \mu_U(t) + c_{q,N} \sigma_U(t) + \varepsilon_{t,\text{sel}}, \end{aligned} \tag{35}$$

with

$$|\varepsilon_{t,\text{rand}}|, |\varepsilon_{t,\text{sel}}| \le \eta_t^2 C_t.$$

Consequently,

$$G_t \;\triangleq\; R_t^{\text{sel}} - R_t^{\text{rand}} = c_{q,N} \sigma_U(t) + \varepsilon_{t,G}, \qquad |\varepsilon_{t,G}| \le 2\eta_t^2 C_t. \tag{36}$$

The population coefficient in Theorem 1 is obtained by letting $N \to \infty$ while keeping $q = B/N \in (0,1)$ fixed. Consistency of upper-trimmed L-statistics gives

$$c_{q,N} \longrightarrow c_q \;\triangleq\; \frac{1}{q} \int_{1-q}^{1} \Phi^{-1}(u)\, \mathrm{d}u. \tag{37}$$

Equivalently, with

$$z_q \triangleq \Phi^{-1}(1-q),$$

continuity of the Gaussian law gives

$$c_q = \mathbb{E}[Z \mid Z \geq z_q]\,. \tag{38}$$

Since $\phi'(z) = -z\phi(z)$,

$$\begin{aligned}\mathbb{E}[Z\mathbf{1}\{Z \geq z_q\}] &= \int_{z_q}^{\infty} z\phi(z)\,\mathrm{d}z \\ &= [-\phi(z)]_{z_q}^{\infty} = \phi(z_q).\end{aligned} \tag{39}$$

Dividing by $\Pr(Z \geq z_q) = q$ yields

$$c_q = \frac{\phi(z_q)}{q}. \tag{40}$$

In particular, when $\lambda = 2$ and hence $q = 1/2$,

$$c_{1/2} = 2\phi(0) = \sqrt{\frac{2}{\pi}} \approx 0.79788. \tag{41}$$

Thus, under the population top-q idealization, (36) becomes

$$G_t = c_q \sigma_U(t) + \mathrm{O}(\eta_t^2 C_t), \tag{42}$$

which is (4). For a finite candidate pool, the mathematically exact statement is (36). Alternatively, invoking the standard fixed-q Gaussian L-statistic bound $c_{q,N} = c_q + \mathrm{O}(N^{-1/2})$ gives

$$G_t = c_q \sigma_U(t) + \mathrm{O}\left(\eta_t^2 C_t + \frac{\sigma_U(t)}{\sqrt{\lambda B}}\right). \tag{43}$$

Because c_q is defined after centering and standardization, it is dimensionless and invariant under every positive affine transformation of the utility scores. □

Corollary 1 (Relative selection gain)**.** *Under the assumptions of Theorem 1, suppose that $\mu_U(t) > 0$ and that the aggregate learning-rate and finite-pool remainder is $o(\mu_U(t))$. Then*

$$\frac{G_t}{R_t^{\mathrm{rand}}} = c_q \frac{\sigma_U(t)}{\mu_U(t)} + o(1) = c_q\,\mathrm{CV}[U(\cdot,t)] + o(1), \tag{44}$$

or, equivalently, $R_t^{\mathrm{sel}}/R_t^{\mathrm{rand}} = 1 + c_q\,\mathrm{CV}[U(\cdot,t)] + o(1)$. Thus, for a fixed retained fraction q, the relative benefit of selection is governed by the coefficient of variation of the optimizer-induced utility scores; the complete finite-pool and remainder bounds are given in Appendix C.1.

Proof of Corollary 1. Write

$$R_t^{\mathrm{rand}} = \mu_U(t) + \varepsilon_{t,\mathrm{rand}}, \qquad G_t = c_{q,N}\sigma_U(t) + \varepsilon_{t,G}, \tag{45}$$

where

$$|\varepsilon_{t,\mathrm{rand}}| \leq \eta_t^2 C_t, \qquad |\varepsilon_{t,G}| \leq 2\eta_t^2 C_t.$$

Suppose that $\mu_U(t) > 0$ and $\eta_t^2 C_t \leq \mu_U(t)/2$. Direct substitution gives

$$\begin{aligned}\frac{G_t}{R_t^{\mathrm{rand}}} &= \frac{c_{q,N}\sigma_U(t) + \varepsilon_{t,G}}{\mu_U(t) + \varepsilon_{t,\mathrm{rand}}} \\ &= c_{q,N}\frac{\sigma_U(t)}{\mu_U(t)} + \mathcal{O}\left(\frac{\eta_t^2 C_t}{\mu_U(t)}\left[1 + c_{q,N}\frac{\sigma_U(t)}{\mu_U(t)}\right]\right).\end{aligned} \tag{46}$$

Hence, to leading order,

$$\frac{G_t}{R_t^{\mathrm{rand}}} = c_{q,N}\,\mathrm{CV}[U(\cdot,t)] + o(1). \tag{47}$$

In the population limit, $c_{q,N} \to c_q$, and therefore

$$\frac{G_t}{R_t^{\mathrm{rand}}} = c_q \,\mathrm{CV}[U(\cdot,t)] + o(1). \tag{48}$$

Equivalently, the multiplicative one-step improvement satisfies

$$\frac{R_t^{\mathrm{sel}}}{R_t^{\mathrm{rand}}} = 1 + c_q \,\mathrm{CV}[U(\cdot,t)] + o(1). \tag{49}$$

□

Remark 2 (Linearity and attribution error)**.** *The preceding loss-reduction argument applies when the same frozen linear map, with matching scale, is used to score candidates and form the batch update. Linearity holds for the specified* `AdamW`*,* `AdEMAMix`*, and unclipped* `Sophia` *scoring maps; agreement with the full optimizer update is a separate assumption.*

For any mismatched optimizer update satisfying the corresponding smoothness and moment conditions, let

$$\boldsymbol{\Delta}_t^a \;\triangleq\; \mathbf{d}_{t,\mathrm{upd}}^a - \frac{1}{B}\sum_{z\in\mathcal{B}_t^a} \mathcal{O}_{t,\mathrm{score}}(\mathbf{g}_{z,t}) \tag{50}$$

denote the discrepancy between the direction actually applied and the additive direction represented by the scoring map. The same proof then acquires the additional error

$$\eta_t \mathbb{E}[\langle \mathbf{g}_{\mathcal{V},t},\, \boldsymbol{\Delta}_t^a \rangle]\,. \tag{51}$$

Thus, the $\mathrm{O}(\eta_t^2)$ *conclusion extends to a nonlinear update only when this attribution error is itself* $\mathrm{O}(\eta_t^2)$*. It is zero for the exact linear families and is controlled to first order for the Fréchet-linearized polar map. For the deployed fixed-sign and right-preconditioned proxies, the theorem describes the gain predicted by the additive scoring geometry unless the corresponding update mismatch is bounded separately, as discussed in Remark 1.*

Remark 3 (Scope of the gain comparison)**.** *Theorem 1 compares selection with uniform sampling for a fixed optimizer and scoring geometry, not absolute convergence across optimizers. For* $\mu_U(t) > 0$*, in the population approximation and with negligible remainder terms, the relative extra one-step reduction is* $c_q \,\mathrm{CV}[U(\cdot,t)]$*; the total improvement factor is* $1 + c_q \,\mathrm{CV}[U(\cdot,t)]$*. Score spread alone does not establish alignment with the validation objective: useful selection requires both spread and ranking fidelity.*

Remark 4 (Temperature and the empirical $2\times$ gain)**.** *Boltzmann sampling without replacement interpolates between top-*B *selection as* $\tau \to 0$ *and uniform sampling as* $\tau \to \infty$*. For nondegenerate scores, let* $\kappa_t(\tau) \in [0,1]$ *denote the fraction of the expected top-*B *score gain retained. In the Gaussian population approximation,*

$$G_t \simeq \kappa_t(\tau) c_q \sigma_U(t).$$

With standardized utilities, κ_t *depends on temperature and distributional shape, not raw score scale alone.*

Assuming approximately constant $\kappa_t \approx \bar{\kappa}$ *and* $\mathrm{CV}[U] \approx \rho$*, and neglecting trajectory changes, a heuristic applied-token speedup is* $1 + \bar{\kappa} c_q \rho$*. For* $q = 1/2$ *and* $\bar{\kappa} \approx 1$*, the* $2\times$ *efficiency reported by Wang et al. (2026) corresponds to* $\rho \approx 1/c_q \approx 1.25$*. Indeed, this is a consistency calculation, not a convergence or compute-efficiency guarantee.*

D Detailed Proofs for *Discriminability* (§4)

This section provides proofs for each theorem in §4 with detailed derivations and supporting assumptions (§D.1–§D.3).

D.1 Sign Collapse (Theorem 2)

Throughout this subsection, we condition on the step-t optimizer state and on the frozen scoring reference $\hat{\mathbf{g}}_{\mathcal{V}}$. Consequently, all expectations, probabilities, and variances are taken only over the candidate gradient

$$\mathbf{g}_z \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \qquad \boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, \dots, \sigma_d^2).$$

Coordinates with $[\hat{\mathbf{g}}_{\mathcal{V}}]_j \neq 0$ but $\sigma_j = 0$ contribute only candidate-independent constants to both scores and therefore contribute zero variance. They may be removed without affecting any discriminability statement. Hence, without loss of generality, we take $\sigma_j > 0$ on $\mathrm{supp}(\hat{\mathbf{g}}_{\mathcal{V}})$. We also fix an arbitrary $\{-1, +1\}$-valued convention for $\mathrm{sign}(0)$; under the preceding non-degeneracy condition, the argument of the exact sign map equals zero with probability zero.

Variance of fixed linear diagonal scores. We first record the variance identity used for the linearized score and, subsequently, for the diagonal-operator comparison of Theorem 3. Importantly, this identity applies only to a candidate-independent *linear* diagonal map; it does not apply directly to the exact nonlinear `Lion` score.

Let $\boldsymbol{\pi} \in \mathbb{R}^d$ be fixed conditional on the optimizer state, and define

$$U_z^{(\boldsymbol{\pi})} \triangleq \eta_t \langle \boldsymbol{\pi} \odot \mathbf{g}_z, \hat{\mathbf{g}}_{\mathcal{V}} \rangle = \eta_t \mathbf{g}_z^\top (\boldsymbol{\pi} \odot \hat{\mathbf{g}}_{\mathcal{V}}). \tag{52}$$

Since only $\mathbf{g}_z$ is random,

$$\begin{aligned} \mathrm{Var}\big[U_z^{(\boldsymbol{\pi})}\big] &= \eta_t^2 \big(\boldsymbol{\pi} \odot \hat{\mathbf{g}}_{\mathcal{V}}\big)^\top \boldsymbol{\Sigma} \big(\boldsymbol{\pi} \odot \hat{\mathbf{g}}_{\mathcal{V}}\big) \\ &= \eta_t^2 \sum_{j=1}^d \pi_j^2 \sigma_j^2 [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2. \end{aligned} \tag{53}$$

In particular,

$$\mathrm{Var}\big[U_z^{\mathrm{SGD}}\big] = \eta_t^2 \sum_{j=1}^d \sigma_j^2 [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2 = \eta_t^2 \left\| \boldsymbol{\Sigma}^{1/2} \hat{\mathbf{g}}_{\mathcal{V}} \right\|_2^2. \tag{54}$$

Whenever $\left\| \boldsymbol{\Sigma}^{1/2} \hat{\mathbf{g}}_{\mathcal{V}} \right\|_2 > 0$, define

$$w_j \triangleq \frac{\sigma_j^2 [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2}{\sum_{k=1}^d \sigma_k^2 [\hat{\mathbf{g}}_{\mathcal{V}}]_k^2}, \qquad \sum_{j=1}^d w_j = 1. \tag{55}$$

Then every comparison among fixed linear diagonal operators satisfies

$$\frac{\mathrm{Var}[U_z^{(\boldsymbol{\pi})}]}{\mathrm{Var}[U_z^{\mathrm{SGD}}]} = \sum_{j=1}^d w_j \pi_j^2. \tag{56}$$

If $\left\| \boldsymbol{\Sigma}^{1/2} \hat{\mathbf{g}}_{\mathcal{V}} \right\|_2 = 0$, the `SGD` and linearized sign scores are both deterministic, so their variance equality remains valid, although the normalized ratio (56) is undefined.

Assumption 1 (Momentum dominance)**.** *Let $0 < \beta_1 < 1$. There exists $\kappa \geq 1$ such that*

$$\beta_1 |[\mathbf{m}_{t-1}]_j| \geq \kappa (1 - \beta_1)\big(\sigma_j + |\mu_j|\big), \qquad \forall j \in \mathrm{supp}(\hat{\mathbf{g}}_{\mathcal{V}}). \tag{57}$$

To interpret the assumption, define the coordinate-wise dominance statistic

$$\kappa_{t,j} \triangleq \frac{\beta_1 |[\mathbf{m}_{t-1}]_j|}{(1 - \beta_1)\big(\sigma_j + |\mu_j|\big)}. \tag{58}$$

Assumption 1 is equivalent to requiring $\kappa_{t,j} \geq \kappa$ on the support of the scoring reference. It is therefore a state-dependent and directly measurable condition, rather than a universal property of sign-based optimization.

As a heuristic interpretation, under stationary i.i.d. gradients the stored momentum buffer approaches the expectation $\mathbb{E}[\mathbf{m}_t] \to \boldsymbol{\mu}$; for `Lion`, this remains true even though the stored buffer uses the separate decay coefficient β_2. Replacing the realized buffer by its stationary mean gives the mean-state approximation

$$\kappa_{t,j} \approx \frac{\beta_1|\mu_j|}{(1-\beta_1)\big(\sigma_j + |\mu_j|\big)}. \tag{59}$$

For $\beta_1 = 0.9$, the prefactor $\beta_1/(1-\beta_1) = 9$ permits a substantial dominance margin when the gradient retains a persistent mean component. This stationary approximation is only interpretive and is not used in the proof below.

Proposition 1 (Linearized sign scoring is spread-neutral)**.** *For every fixed sign pattern $\mathbf{s}_t \in \{-1,+1\}^d$, the linearized sign score*

$$U_z^{\mathtt{Lion},\mathrm{lin}} \triangleq \eta_t \langle \mathbf{g}_z,\, \mathbf{s}_t \odot \hat{\mathbf{g}}_{\mathcal{V}} \rangle \tag{60}$$

satisfies

$$\mathrm{Var}\big[U_z^{\mathtt{Lion},\mathrm{lin}}\big] = \eta_t^2 \sum_{j=1}^{d} \sigma_j^2 [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2 = \eta_t^2 \left\| \boldsymbol{\Sigma}^{1/2} \hat{\mathbf{g}}_{\mathcal{V}} \right\|_2^2 = \mathrm{Var}\big[U_z^{\mathtt{SGD}}\big]. \tag{5}$$

Proof of Proposition 1. The linearized score is a fixed linear functional of the candidate:

$$U_z^{\mathrm{Lion,lin}} = \eta_t \sum_{j=1}^{d} [\mathbf{g}_z]_j [\mathbf{s}_t]_j [\hat{\mathbf{g}}_{\mathcal{V}}]_j. \tag{61}$$

Applying the fixed-diagonal variance identity (53) with $\pi_j = [\mathbf{s}_t]_j$ gives

$$\begin{aligned} \mathrm{Var}\big[U_z^{\mathrm{Lion,lin}}\big] &= \eta_t^2 \sum_{j=1}^{d} [\mathbf{s}_t]_j^2 \sigma_j^2 [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2 \\ &= \eta_t^2 \sum_{j=1}^{d} \sigma_j^2 [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2, \end{aligned} \tag{62}$$

because $[\mathbf{s}_t]_j^2 = 1$ for every coordinate. This is exactly the `SGD` score variance. Equivalently, the diagonal reflection $\mathrm{diag}(\mathbf{s}_t)$ preserves the covariance geometry because

$$\mathrm{diag}(\mathbf{s}_t)\boldsymbol{\Sigma}\mathrm{diag}(\mathbf{s}_t) = \boldsymbol{\Sigma}.$$

□

Proposition 2 (Exponential collapse of the exact score)**.** *Suppose that, for some $\kappa \geq 1$,*

$$\beta_1 |[\mathbf{m}_{t-1}]_j| \;\geq\; \kappa(1-\beta_1)\big(\sigma_j + |\mu_j|\big), \qquad \forall j \in \mathrm{supp}(\hat{\mathbf{g}}_{\mathcal{V}}). \tag{63}$$

Then the exact `Lion` score

$$U_z^{\mathtt{Lion}} \triangleq \eta_t \langle \mathrm{sign}\big(\beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_z\big),\, \hat{\mathbf{g}}_{\mathcal{V}} \rangle \tag{64}$$

satisfies

$$\mathrm{Var}\big[U_z^{\mathtt{Lion}}\big] \;\leq\; 8e^{-\kappa^2/2} \eta_t^2 \left\| \hat{\mathbf{g}}_{\mathcal{V}} \right\|_2^2. \tag{65}$$

Consequently, whenever $\left\| \boldsymbol{\Sigma}^{1/2} \hat{\mathbf{g}}_{\mathcal{V}} \right\|_2 > 0$,

$$\mathrm{Var}\big[U_z^{\mathtt{Lion}}\big] \;\leq\; \mathrm{Var}\big[U_z^{\mathtt{Lion,lin}}\big] = \mathrm{Var}\big[U_z^{\mathtt{SGD}}\big] \quad \textit{if} \quad \kappa^2 \;\geq\; 2\log\left(\frac{8\left\| \hat{\mathbf{g}}_{\mathcal{V}} \right\|_2^2}{\left\| \boldsymbol{\Sigma}^{1/2} \hat{\mathbf{g}}_{\mathcal{V}} \right\|_2^2} \right). \tag{66}$$

The same conclusions hold for `Signum` after replacing β_1 with its momentum coefficient μ.

Proof of Proposition 2. Step 1: define coordinate-wise sign flips

We first apply exact decomposition into coordinate-wise sign flips. Denote

$$[\mathbf{s}_t]_j \triangleq \operatorname{sign}([\mathbf{m}_{t-1}]_j), \tag{67}$$

and the exact candidate-dependent sign

$$\xi_{j,z} \triangleq \operatorname{sign}(\beta_1[\mathbf{m}_{t-1}]_j + (1-\beta_1)[\mathbf{g}_z]_j)\,. \tag{68}$$

For each coordinate, introduce the flip indicator, which is the probability that the candidate gradient flips the sign of the momentum buffer at j^{th} coordinate.

$$F_{j,z} \triangleq \mathbf{1}\{\xi_{j,z} \neq [\mathbf{s}_t]_j\}, \qquad p_j \triangleq \Pr_z(F_{j,z}=1). \tag{69}$$

Since both $\xi_{j,z}$ and $[\mathbf{s}_t]_j$ take values in $\{-1,+1\}$,

$$\mathbb{E}[\xi_{j,z}] = [\mathbf{s}_t]_j\big(1-2p_j\big), \mathbb{E}[\xi_{j,z}^2] = 1. \tag{70}$$

Therefore, we have

$$\operatorname{Var}[\xi_{j,z}] = 1-(1-2p_j)^2 = 4p_j(1-p_j) \le 4p_j. \tag{71}$$

According to our Gaussian candidate modelling, $\mathbf{\Sigma}$ is diagonal and $\mathbf{g}_z$ is Gaussian, so the coordinates $\{[\mathbf{g}_z]_j\}_{j=1}^d$ are independent. Therefore, the flip indicators are independent across coordinates. We can write the variance of the exact score as:

$$\begin{aligned}
\operatorname{Var}\big[U_z^{\texttt{Lion}}\big] &= \operatorname{Var}[\eta_t \sum_{j=1}^d \xi_{j,z}[\hat{\mathbf{g}}_{\mathcal{V}}]_j] \\
&= \eta_t^2 \sum_{j=1}^d [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2 \operatorname{Var}[\xi_{j,z}] \\
&= 4\eta_t^2 \sum_{j=1}^d [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2 p_j(1-p_j) \le 4\eta_t^2 \sum_{j=1}^d [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2 p_j.
\end{aligned} \tag{72}$$

Thus, we find that the candidate-dependent variation in the nonlinear sign score is carried by coordinate-wise sign reversals.

Step 2: sign flips is a Gaussian tail events under momentum dominance.

$\forall j \in \operatorname{supp}(\hat{\mathbf{g}}_{\mathcal{V}})$, a flip occurs when the candidate gradient satisfies:

$$\{F_{j,z}=1\} = \{[\mathbf{s}_t]_j[\mathbf{g}_z]_j < -a_{t,j}\}\,, a_{t,j} \triangleq \frac{\beta_1|[\mathbf{m}_{t-1}]_j|}{1-\beta_1}. \tag{73}$$

Since

$$[\mathbf{s}_t]_j[\mathbf{g}_z]_j \sim \mathcal{N}\big([\mathbf{s}_t]_j\mu_j, \sigma_j^2\big)\,,$$

the flip probability can be written in a Gaussian cumulative distribution function $\Phi(\cdot)$:

$$p_j = \mathbf{P}[\{[\mathbf{s}_t]_j[\mathbf{g}_z]_j < -a_{t,j}\}] = \Phi\left(\frac{-a_{t,j}-[\mathbf{s}_t]_j\mu_j}{\sigma_j}\right). \tag{74}$$

According to Assumption 1, we have $a_{t,j} \ge \kappa\big(\sigma_j + |\mu_j|\big)$, therefore,

$$\frac{-a_{t,j}-[\mathbf{s}_t]_j\mu_j}{\sigma_j} \le -\kappa - \frac{\kappa|\mu_j| + [\mathbf{s}_t]_j\mu_j}{\sigma_j} \le -\kappa. \tag{75}$$

The second inequality is from $\kappa|\mu_j| + [\mathbf{s}_t]_j\mu_j \ge (\kappa-1)|\mu_j| \ge 0$.

Combining (74) and (75) gives the sharper one-sided bound

$$p_j \le \Phi(-\kappa). \tag{76}$$

In particular, for $Z \sim \mathcal{N}(0,1)$, according to Gaussian Chernoff tail bound, we have

$$p_j \le \Phi(-\kappa) \le \Pr(|Z| \ge \kappa) \le 2e^{-\kappa^2/2}. \tag{77}$$

Substituting (77) into (72), we obtain

$$\begin{aligned}\mathrm{Var}\big[U_z^{\mathtt{Lion}}\big] &\le 4\eta_t^2 \sum_{j=1}^{d} [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2 p_j \\ &\le 8e^{-\kappa^2/2}\eta_t^2 \sum_{j=1}^{d} [\hat{\mathbf{g}}_{\mathcal{V}}]_j^2 = 8e^{-\kappa^2/2}\eta_t^2 \left\|\hat{\mathbf{g}}_{\mathcal{V}}\right\|_2^2, \end{aligned} \tag{78}$$

which proves Eq. (65).

Finally, when $\left\|\boldsymbol{\Sigma}^{1/2}\hat{\mathbf{g}}_{\mathcal{V}}\right\|_2 > 0$, combining Eqs. (5) and (65) yields

$$\frac{\mathrm{Var}[U_z^{\mathtt{Lion}}]}{\mathrm{Var}[U_z^{\mathtt{Lion,lin}}]} \le 8e^{-\kappa^2/2} \frac{\left\|\hat{\mathbf{g}}_{\mathcal{V}}\right\|_2^2}{\left\|\boldsymbol{\Sigma}^{1/2}\hat{\mathbf{g}}_{\mathcal{V}}\right\|_2^2}. \tag{79}$$

Therefore,

$$\mathrm{Var}\big[U_z^{\mathtt{Lion}}\big] \le \mathrm{Var}\big[U_z^{\mathtt{Lion,lin}}\big] \tag{80}$$

whenever

$$\kappa^2 \ge 2\log\left(\frac{8\left\|\hat{\mathbf{g}}_{\mathcal{V}}\right\|_2^2}{\left\|\boldsymbol{\Sigma}^{1/2}\hat{\mathbf{g}}_{\mathcal{V}}\right\|_2^2}\right). \tag{81}$$

The proof uses only the generic sign-operator form

$$\mathrm{sign}(\alpha\mathbf{m}_{t-1} + (1-\alpha)\mathbf{g}_z)\,.$$

The same proof applies to `Signum` by replacing β_1 with Signum's scalar momentum coefficient μ, which is distinct from the gradient-mean vector $\boldsymbol{\mu}$. □

The two parts of Theorem 2 concern distinct scoring rules. Proposition 1 analyzes the fixed-sign linear score denoted by $U_z^{\mathtt{Lion,lin}}$, whereas (65) analyzes the exact nonlinear score $U_z^{\mathtt{Lion}}$. Accordingly, the fixed-diagonal spread identity (56) may be applied to the former but not to the latter. Conversely, the momentum-dominance bound for the exact score does not by itself constitute a variance bound for the implemented linearized score.

D.2 Diagonal-Adaptive Advantage of AdamW and Sophia (Theorem 3)

The amplification half of the ranking rests on one further empirical condition and one convexity lemma, from which the two auxiliary propositions and the theorem follow.

Assumption 2 (Weighted variance normalization)**.** *With the weights w_j of (55), the second-moment estimates satisfy $\sum_j w_j\,[\hat{\mathbf{v}}_t]_j \le 1$, and $[\hat{\mathbf{v}}_t]_j$ is non-constant on the support of w.*

Assumption 2 asks that the coordinates carrying validation-relevant variance not sit where the second moments are large. It is empirically mild since gradient clipping caps the incoming squared gradients, and the downstream proxy concentrates w on rare-feature coordinates, precisely those with $[\hat{\mathbf{v}}_t]_j \ll 1$. We start the comparison on `AdamW` and `Sophia` from Lemma 2:

Lemma 2 (Preconditioner amplification)**.** *Let $w_j > 0$ be weights with $\sum_j w_j = 1$, and let $c_j > 0$ be positive scalars that are not all equal. Then $\sum_j w_j\, c_j^2 > \left(\sum_j w_j\, c_j\right)^2$. Moreover, for $c_j = [\mathbf{c}_t]_j = 1/(\sqrt{[\hat{\mathbf{v}}_t]_j} + \varepsilon)$ under Assumption 2,*

$$\sum_j w_j\,[\mathbf{c}_t]_j^2 \;>\; \frac{1}{(1+\varepsilon')^2} \qquad \textit{with} \quad \varepsilon' \triangleq \frac{\varepsilon}{\min_j \sqrt{[\hat{\mathbf{v}}_t]_j}},$$

and in the regime $\varepsilon \ll \min_j \sqrt{[\hat{\mathbf{v}}_t]_j}$ relevant to `AdamW` ($\varepsilon = 10^{-8}$),

$$\sum_j w_j\,[\mathbf{c}_t]_j^2 \;>\; 1 - \mathrm{O}(\varepsilon').$$

Proof of Lemma 2 (Preconditioner amplification). The first statement is an immediate consequence of Jensen's inequality applied to the strictly convex function $\varphi(x) = x^2$: $\mathbb{E}_w[\varphi(c_J)] > \varphi(\mathbb{E}_w[c_J])$ when c_J is non-constant under the distribution w.

For the second statement, write $\sqrt{[\hat{\mathbf{v}}_t]_j} + \varepsilon = \sqrt{[\hat{\mathbf{v}}_t]_j}\,(1 + \varepsilon/\sqrt{[\hat{\mathbf{v}}_t]_j}) \le \sqrt{[\hat{\mathbf{v}}_t]_j}\,(1+\varepsilon')$, so that $[\mathbf{c}_t]_j^2 \ge \frac{1}{(1+\varepsilon')^2} \cdot \frac{1}{[\hat{\mathbf{v}}_t]_j}$ coordinate-wise; it therefore suffices to bound $\sum_j w_j/[\hat{\mathbf{v}}_t]_j$. By Jensen's inequality applied to the convex function $x \mapsto 1/x$ on $\mathbb{R}_{>0}$,

$$\sum_j \frac{w_j}{[\hat{\mathbf{v}}_t]_j} \;\ge\; \frac{1}{\sum_j w_j\,[\hat{\mathbf{v}}_t]_j} \;\ge\; 1,$$

where the last step uses Assumption 2; the inequality is strict since $[\hat{\mathbf{v}}_t]_j$ is non-constant on the support of w. The $\varepsilon \to 0$ statement follows since $\varepsilon' = \mathcal{O}(\varepsilon)$ when $\varepsilon \ll \min_j \sqrt{[\hat{\mathbf{v}}_t]_j}$. We also discuss the role of the normalization condition; without it the conclusion can fail (e.g., $w = (\frac{1}{2}, \frac{1}{2})$, $\hat{\mathbf{v}}_t = (9,\, 0.81)$ gives $\sum_j w_j[\mathbf{c}_t]_j^2 \approx 0.67$), so Assumption 2 is not merely technical. □

Proposition 3 (Variance preconditioning amplifies spread)**.** *The* `AdamW` *score* $U_z^{\texttt{Adam}} = \eta_t \langle \mathbf{g}_z,\, \mathbf{c}_t \odot \mathbf{g}_{\mathcal{V}} \rangle$ *has*

$$\mathrm{Var}[U_z^{\texttt{Adam}}] \;=\; \eta_t^2 \sum_{j=1}^d \sigma_j^2\,[\mathbf{c}_t]_j^2\,[\mathbf{g}_{\mathcal{V}}]_j^2 \;=\; \eta_t^2 \left\| \mathbf{\Sigma}^{1/2} (\mathbf{c}_t \odot \mathbf{g}_{\mathcal{V}}) \right\|_2^2, \tag{82}$$

so its spread ratio (56) *is* $\sum_j w_j\,[\mathbf{c}_t]_j^2$*, which under Assumption 2, Lemma 2 bounds below:*

$$\frac{\mathrm{Var}[U_z^{\texttt{Adam}}]}{\mathrm{Var}[U_z^{\texttt{SGD}}]} \;=\; \sum_{j=1}^d w_j\,[\mathbf{c}_t]_j^2 \;>\; 1 - \mathrm{O}(\varepsilon').$$

Proof of Proposition 3 (Variance preconditioning amplifies spread). The `Adam` influence score is $U_z^{\texttt{Adam}} = \eta_t \langle \mathbf{c}_t \odot \mathbf{g}_z,\, \mathbf{g}_{\mathcal{V}} \rangle = \eta_t \sum_j [\mathbf{c}_t]_j [\mathbf{g}_z]_j [\mathbf{g}_{\mathcal{V}}]_j$. By independence of the coordinates of $\mathbf{g}_z$: $\mathrm{Var}[U_z^{\texttt{Adam}}] = \eta_t^2 \sum_j [\mathbf{c}_t]_j^2\,[\mathbf{g}_{\mathcal{V}}]_j^2\,\sigma_j^2$, yielding (82). Dividing by $\mathrm{Var}[U_z^{\texttt{SGD}}] = \eta_t^2 \sum_j \sigma_j^2 [\mathbf{g}_{\mathcal{V}}]_j^2$ gives the spread ratio $\sum_j w_j [\mathbf{c}_t]_j^2$ with the weights (55), and Lemma 2 under Assumption 2 bounds it below by $1 - \mathrm{O}(\varepsilon')$. □

Proposition 4 (Curvature preconditioning amplifies spread)**.** *The* `Sophia` *score under the unclipped scoring operator* (13) *is the diagonal-form score with pattern* $\pi_j^{\texttt{Sophia}} = 1/\max\{\gamma[\mathbf{h}_t]_j,\, \varepsilon\}$*. Suppose the curvature estimates satisfy the normalization analogous to Assumption 2:* $\sum_j w_j\,\gamma[\mathbf{h}_t]_j \le 1$*, with* $[\mathbf{h}_t]_j$ *non-constant on the support of* w*. Then, with* $\varepsilon'' \triangleq \varepsilon/\min_j \gamma[\mathbf{h}_t]_j$*,*

$$\frac{\mathrm{Var}[U_z^{\texttt{Sophia}}]}{\mathrm{Var}[U_z^{\texttt{SGD}}]} \;=\; \sum_{j=1}^d w_j \left(\pi_j^{\texttt{Sophia}}\right)^2 \;>\; \frac{1}{(1+\varepsilon'')^2} \;=\; 1 - \mathrm{O}(\varepsilon'')$$

in the regime $\varepsilon \ll \min_j \gamma[\mathbf{h}_t]_j$*. The amplification enters through the* squared *inverse curvature, concentrated on exactly the flat directions (small* $[\mathbf{h}_t]_j$*) that the validation-relevant weights* w_j *occupy; the condition is measurable along training in the same way as Assumption 2.*

Proof. Since $\max\{\gamma[\mathbf{h}_t]_j, \varepsilon\} \le \gamma[\mathbf{h}_t]_j\,(1 + \varepsilon/\gamma[\mathbf{h}_t]_j) \le \gamma[\mathbf{h}_t]_j\,(1+\varepsilon'')$, we have $\left(\pi_j^{\texttt{Sophia}}\right)^2 \ge (1+\varepsilon'')^{-2}\,(\gamma[\mathbf{h}_t]_j)^{-2}$ coordinate-wise. Applying Jensen's inequality twice, to $x \mapsto x^2$ (strictly, since $[\mathbf{h}_t]_j$ is non-constant on the support of w) and to the convex map $x \mapsto 1/x$,

$$\sum_j w_j\,(\gamma[\mathbf{h}_t]_j)^{-2} \;>\; \Big(\sum_j w_j\,(\gamma[\mathbf{h}_t]_j)^{-1}\Big)^2 \;\ge\; \Big(\sum_j w_j\,\gamma[\mathbf{h}_t]_j\Big)^{-2} \;\ge\; 1,$$

where the last step uses the normalization. Combining the two displays gives the claim; the spread-ratio identity is (56). □

Combining Proposition 3–4 with Proposition 2 (66), we complete the proof of Theorem 3.

D.3 Radial Collapse of Muon (Theorem 4)

D.3.1 Preliminaries: the ideal polar map

Throughout, all matrices are real, and msign denotes the ideal polar-factor map, not the conventional primary matrix-sign function. The exact identities below do not describe the finite Newton–Schulz computation in (16). We condition on the momentum matrix $\mathbf{A}_t$, the scoring reference $\mathbf{R} = \hat{\mathbf{G}}_{\mathcal{V}} \in \mathbb{R}^{m\times n}$, and all scoring coefficients. All probabilistic statements are conditional on these frozen quantities; only the candidate gradient $\mathbf{G}_z$ is random.

Assumption 3 (Regular polar state)**.** *The matrix $\mathbf{A}_t \in \mathbb{R}^{m\times n}$, $m \geq n \geq 1$, has full column rank. Fix a compact SVD*

$$\mathbf{A}_t = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \qquad \mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_n), \qquad \sigma_1 \geq \cdots \geq \sigma_n > 0, \tag{83}$$

where $\mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}_n$ and $\mathbf{V}$ is square. Repeated singular values are allowed.

The polar decomposition is $\mathbf{A}_t = \mathbf{Q}\mathbf{H}$, where

$$\mathbf{Q} = \mathrm{msign}(\mathbf{A}_t) = \mathbf{A}_t(\mathbf{A}_t^\top\mathbf{A}_t)^{-1/2} = \mathbf{U}\mathbf{V}^\top, \qquad \mathbf{H} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^\top \succ 0. \tag{84}$$

In particular, $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_n$. The polar-derivative statements for full-row-rank wide matrices follow by transposition. The right-preconditioned proxy is analyzed below in the displayed $m \geq n$ orientation.

Proposition 5 (Spectral-norm steepest descent)**.** *For full-column-rank $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, $\mathrm{msign}(\mathbf{A}) = \mathbf{U}\mathbf{V}^\top$ is the unique maximizer of $\langle \mathbf{A}, \mathbf{D}\rangle_F$ over $\|\mathbf{D}\|_{\mathrm{op}} \leq 1$.*

Proof. For any feasible $\mathbf{D}$, its alignment with the singular vectors satisfies

$$\langle \mathbf{A}, \mathbf{D}\rangle_F = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i^\top \mathbf{D}\mathbf{v}_i \leq \sum_{i=1}^{n} \sigma_i,$$

with equality at $\mathbf{D} = \mathbf{U}\mathbf{V}^\top$. Since every $\sigma_i > 0$, equality requires $\mathbf{u}_i^\top\mathbf{D}\mathbf{v}_i = 1$ for every i. Together with $\|\mathbf{D}\mathbf{v}_i\|_2 \leq 1$, this implies $\mathbf{D}\mathbf{v}_i = \mathbf{u}_i$. The vectors $\mathbf{v}_i$ form a basis, so $\mathbf{D} = \mathbf{U}\mathbf{V}^\top$. □

D.3.2 Polar derivative and radial geometry

Fix an orthonormal complement $\mathbf{U}_\perp \in \mathbb{R}^{m\times(m-n)}$ so that $[\mathbf{U}\ \mathbf{U}_\perp]$ is orthogonal. Define

$$\mathbf{B}_z = \mathbf{U}^\top\mathbf{G}_z\mathbf{V}, \qquad \mathbf{C}_z = \mathbf{U}_\perp^\top\mathbf{G}_z\mathbf{V}, \tag{85}$$

$$\mathbf{R}_\parallel = \mathbf{U}^\top\mathbf{R}\mathbf{V}, \qquad \mathbf{R}_\perp = \mathbf{U}_\perp^\top\mathbf{R}\mathbf{V}. \tag{86}$$

When $m = n$, complement-block terms are absent. Unless indicated otherwise, i, j range over $1, \ldots, n$ and a over $1, \ldots, m-n$.

The radial subspace and its Frobenius-orthogonal complement are

$$\mathcal{R}_t = \{\mathbf{U}\mathbf{S}\mathbf{V}^\top : \mathbf{S} \in \mathbb{R}^{n\times n},\ \mathbf{S} = \mathbf{S}^\top\},$$
$$\mathcal{R}_t^\perp = \{\mathbf{X} \in \mathbb{R}^{m\times n} : \mathbf{Q}^\top\mathbf{X} + \mathbf{X}^\top\mathbf{Q} = \mathbf{0}\}.$$

Equivalently, $\mathcal{R}_t = \{\mathbf{Q}\mathbf{S} : \mathbf{S} = \mathbf{S}^\top\}$: radial perturbations change the positive polar factor at fixed $\mathbf{Q}$. Writing $\mathrm{sym}(\mathbf{X}) = (\mathbf{X} + \mathbf{X}^\top)/2$, define

$$\Pi^{\mathrm{rad}}\mathbf{M} = \mathbf{U}\,\mathrm{sym}(\mathbf{U}^\top\mathbf{M}\mathbf{V})\mathbf{V}^\top, \qquad \Pi^{\tan} = \mathrm{Id} - \Pi^{\mathrm{rad}}. \tag{87}$$

These are the orthogonal projections onto $\mathcal{R}_t$ and $\mathcal{R}_t^\perp$, respectively, since symmetric and skew-symmetric matrices are Frobenius-orthogonal and the two column-space blocks are orthogonal. Set $\mathbf{G}_z^a = \Pi^a\mathbf{G}_z$ for $a \in \{\mathrm{rad}, \tan\}$.

Theorem 5 (Exact first-order geometry of matrix-sign scoring)**.** *Under Assumption 3, let $\mathcal{L}_t = \mathrm{D}\,\mathrm{msign}(\mathbf{A}_t)$. For any $\mathbf{G} \in \mathbb{R}^{m\times n}$, set $\mathbf{B} = \mathbf{U}^\top\mathbf{G}\mathbf{V}$ and $\mathbf{C} = \mathbf{U}_\perp^\top\mathbf{G}\mathbf{V}$. Then*

$$\mathcal{L}_t[\mathbf{G}] = \mathbf{U}\mathbf{\Omega}(\mathbf{B})\mathbf{V}^\top + \mathbf{U}_\perp\mathbf{C}\mathbf{\Sigma}^{-1}\mathbf{V}^\top,$$
$$[\mathbf{\Omega}(\mathbf{B})]_{ij} = \frac{[\mathbf{B}]_{ij} - [\mathbf{B}]_{ji}}{\sigma_i + \sigma_j}, \qquad 1 \leq i, j \leq n. \tag{88}$$

The derivative is self-adjoint for the Frobenius inner product:

$$\mathcal{T}_{\mathbf{A}_t} \triangleq \mathcal{L}_t^* = \mathcal{L}_t. \tag{89}$$

Moreover, with $\delta_t = \sigma_n/2$, there exists $C_t < \infty$, depending only on $\mathbf{A}_t$, such that

$$\begin{aligned} \Phi_z(\alpha) &\triangleq \langle \mathrm{msign}(\mathbf{A}_t + \alpha \mathbf{G}_z),\, \mathbf{R}\rangle_F \\ &= \langle \mathbf{Q},\, \mathbf{R}\rangle_F + \alpha \langle \mathbf{G}_z,\, \mathcal{L}_t[\mathbf{R}]\rangle_F + r_z(\alpha), \\ |r_z(\alpha)| &\le C_t \,\|\mathbf{R}\|_F\, \alpha^2\, \|\mathbf{G}_z\|_F^2 \end{aligned} \tag{90}$$

for every real α and every $\mathbf{G}_z$ satisfying $|\alpha|\,\|\mathbf{G}_z\|_F \le \delta_t$.

Proof. The polar map is smooth on the open set of full-column-rank matrices. Differentiate $\mathbf{A}_t = \mathbf{Q}\mathbf{H}$ in direction $\mathbf{G}$ to obtain $\mathbf{G} = \dot{\mathbf{Q}}\mathbf{H} + \mathbf{Q}\dot{\mathbf{H}}$. Since $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_n$ and $\dot{\mathbf{H}}$ is symmetric, $\mathbf{K} = \mathbf{Q}^\top\dot{\mathbf{Q}}$ is skew-symmetric, and

$$\begin{aligned} \mathbf{K}\mathbf{H} + \mathbf{H}\mathbf{K} &= \mathbf{Q}^\top\mathbf{G} - \mathbf{G}^\top\mathbf{Q}, \\ (\mathbf{I}_m - \mathbf{Q}\mathbf{Q}^\top)\dot{\mathbf{Q}} &= (\mathbf{I}_m - \mathbf{Q}\mathbf{Q}^\top)\mathbf{G}\mathbf{H}^{-1}. \end{aligned}$$

In the $\mathbf{V}$ basis, the first identity becomes $\mathbf{\Omega}\mathbf{\Sigma} + \mathbf{\Sigma}\mathbf{\Omega} = \mathbf{B} - \mathbf{B}^\top$, where $\mathbf{\Omega} = \mathbf{V}^\top\mathbf{K}\mathbf{V}$. Solving entrywise and using $\dot{\mathbf{Q}} = \mathbf{Q}\mathbf{K} + (\mathbf{I}_m - \mathbf{Q}\mathbf{Q}^\top)\dot{\mathbf{Q}}$ gives (88).

In the orthogonal block coordinates $(\mathbf{B}, \mathbf{C})$, the map on each pair $([\mathbf{B}]_{ij}, [\mathbf{B}]_{ji})$, $i < j$, has the symmetric matrix

$$\frac{1}{\sigma_i + \sigma_j}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix};$$

diagonal entries are sent to zero, and each $[\mathbf{C}]_{aj}$ is multiplied by σ_j^{-1}. Thus $\mathcal{L}_t$ is self-adjoint. Finally, the closed Frobenius ball of radius $\delta_t = \sigma_n/2$ around $\mathbf{A}_t$ lies in the full-column-rank set. The second derivative is bounded on this compact ball, so Taylor's theorem and Cauchy–Schwarz give (90). □

Corollary 2 (Exact radial nullspace)**.** *Under Assumption 3,*

$$\ker \mathcal{L}_t = \mathcal{R}_t, \qquad \mathcal{L}_t = \mathcal{L}_t\Pi^{\tan} = \Pi^{\tan}\mathcal{L}_t.$$

Proof. The two terms in (88) are orthogonal and vanish simultaneously exactly when $\mathbf{B} = \mathbf{B}^\top$ and $\mathbf{C} = 0$. This gives the kernel and the first projection identity; self-adjointness gives the second. □

Corollary 3 (Exact radial invariance)**.** *Under Assumption 3, for every symmetric $\mathbf{S} \in \mathbb{R}^{n\times n}$ and every real α,*

$$\mathrm{msign}(\mathbf{A}_t + \alpha\mathbf{U}\mathbf{S}\mathbf{V}^\top) = \mathbf{U}\mathbf{V}^\top \quad \textit{whenever} \quad \mathbf{\Sigma} + \alpha\mathbf{S} \succ 0. \tag{91}$$

Proof. The factorization $\mathbf{A}_t + \alpha\mathbf{U}\mathbf{S}\mathbf{V}^\top = (\mathbf{U}\mathbf{V}^\top)[\mathbf{V}(\mathbf{\Sigma} + \alpha\mathbf{S})\mathbf{V}^\top]$ is its polar decomposition. □

A diagonal $\mathbf{S}$ changes only singular values in the chosen basis. A general symmetric $\mathbf{S}$ can also change the eigenvectors of the positive polar factor while leaving $\mathbf{Q}$ unchanged. The finite invariance above concerns purely radial perturbations; for arbitrary candidates differing radially, equality is guaranteed only for their first-order scores at the frozen state.

D.3.3 Score variance and proof of Theorem 4

Fix $\eta_t > 0$ and $\alpha \ge 0$, and assume $\mathbb{E}\,\|\mathbf{G}_z\|_F^2 < \infty$. Define

$$\begin{aligned} U_z^{\texttt{Muon}} &= \eta_t\alpha\langle \mathcal{L}_t[\mathbf{G}_z],\, \mathbf{R}\rangle_F = \eta_t\alpha\langle \mathbf{G}_z,\, \mathcal{L}_t[\mathbf{R}]\rangle_F, \\ U_z^{a} &= \eta_t\langle \mathbf{G}_z^{a},\, \mathbf{R}\rangle_F, \qquad a \in \{\mathrm{rad}, \tan\}, \\ U_z^{\texttt{SGD}} &= \eta_t\langle \mathbf{G}_z,\, \mathbf{R}\rangle_F = U_z^{\mathrm{rad}} + U_z^{\tan}. \end{aligned}$$

Thus $U_z^{\texttt{Muon}}$ is the candidate-dependent first-order term of $\eta_t\Phi_z(\alpha)$.

Corollary 4 (General covariance formula)**.** *Under the frozen-state and second-moment conditions above, let* $\mathbf{\Gamma} = \mathrm{Cov}(\mathrm{vec}\,\mathbf{G}_z)$ *and* $\mathbf{t} = \mathrm{vec}(\mathcal{L}_t[\mathbf{R}])$*. Then*

$$\mathrm{Var}(U_z^{\mathit{Muon}}) = \eta_t^2 \alpha^2 \mathbf{t}^\top \mathbf{\Gamma} \mathbf{t}. \tag{92}$$

No coordinate-uncorrelation assumption is needed.

Proof. Apply the covariance identity for the linear functional $U_z^{\mathtt{Muon}} = \eta_t \alpha\, \mathbf{t}^\top \mathrm{vec}(\mathbf{G}_z)$. □

For the coordinate formulas and Theorem 4, additionally assume that the entries of the combined collection $(\mathbf{B}_z, \mathbf{C}_z)$ are pairwise uncorrelated, including cross-block pairs, and write

$$\tau_{ij}^2 = \mathrm{Var}([\mathbf{B}_z]_{ij}), \qquad \nu_{aj}^2 = \mathrm{Var}([\mathbf{C}_z]_{aj}).$$

These are entry variances, not variances of antisymmetric differences. Uncorrelation is assumed in the fixed singular-vector basis and does not follow merely from diagonal covariance in the original parameter coordinates. Neither Gaussianity nor a zero-mean assumption is required.

Proof of Theorem 4. Corollary 2 gives $\mathcal{L}_t[\mathbf{G}_z] = \mathcal{L}_t[\mathbf{G}_z^{\tan}]$, so changing only the radial component leaves the score unchanged. By self-adjointness and (88),

$$U_z^{\mathtt{Muon}} = \eta_t \alpha \left[\sum_{i \neq j} [\mathbf{B}_z]_{ij} \frac{[\mathbf{R}_\parallel]_{ij} - [\mathbf{R}_\parallel]_{ji}}{\sigma_i + \sigma_j} + \sum_{a,j} [\mathbf{C}_z]_{aj} \frac{[\mathbf{R}_\perp]_{aj}}{\sigma_j} \right].$$

Pairwise uncorrelation yields

$$\mathrm{Var}(U_z^{\mathtt{Muon}}) = \eta_t^2 \alpha^2 \left[\sum_{i \neq j} \tau_{ij}^2 \left(\frac{[\mathbf{R}_\parallel]_{ij} - [\mathbf{R}_\parallel]_{ji}}{\sigma_i + \sigma_j} \right)^2 + \sum_{a,j} \nu_{aj}^2 \left(\frac{[\mathbf{R}_\perp]_{aj}}{\sigma_j} \right)^2 \right]. \tag{93}$$

Likewise, the tangential projection gives

$$\mathrm{Var}(U_z^{\tan}) = \eta_t^2 \left[\sum_{i \neq j} \tau_{ij}^2 \left(\frac{[\mathbf{R}_\parallel]_{ij} - [\mathbf{R}_\parallel]_{ji}}{2} \right)^2 + \sum_{a,j} \nu_{aj}^2 [\mathbf{R}_\perp]_{aj}^2 \right].$$

Since $2/(\sigma_i + \sigma_j) \leq 1/\sigma_n$ and $1/\sigma_j \leq 1/\sigma_n$, termwise comparison gives

$$\mathrm{Var}(U_z^{\mathtt{Muon}}) \leq \frac{\alpha^2}{\sigma_n^2} \mathrm{Var}(U_z^{\tan}),$$

which is (9). □

Comparison with `SGD`. Orthogonality of the matrix projections does not imply uncorrelation of the projected scalar scores. Indeed,

$$\mathrm{Var}(U_z^{\mathtt{SGD}}) = \mathrm{Var}(U_z^{\tan}) + \mathrm{Var}(U_z^{\mathrm{rad}}) + 2\mathrm{Cov}(U_z^{\tan}, U_z^{\mathrm{rad}}),$$

and expansion under the coordinate assumptions above gives

$$\mathrm{Cov}(U_z^{\tan}, U_z^{\mathrm{rad}}) = \frac{\eta_t^2}{4} \sum_{i<j} (\tau_{ij}^2 - \tau_{ji}^2)\big([\mathbf{R}_\parallel]_{ij}^2 - [\mathbf{R}_\parallel]_{ji}^2\big).$$

For example, transpose-pair variance symmetry $\tau_{ij}^2 = \tau_{ji}^2$ makes this covariance zero; together with $\alpha \leq \sigma_n$, it is sufficient for $\mathrm{Var}(U_z^{\mathtt{Muon}}) \leq \mathrm{Var}(U_z^{\mathtt{SGD}})$. There is no unconditional ordering: for $\mathbf{A}_t = s\mathbf{I}_2$, $s > 0$, a nonzero skew-symmetric reference $\mathbf{R}$, and independent unit-variance entries of $\mathbf{G}_z$, the ratio is α^2/s^2, which exceeds one when $0 < s < \alpha$.

D.4 RIGHT-PRECONDITIONED MUON PROXY AND EXACT ERROR

For fixed $\varepsilon \geq 0$, write

$$\mathbf{P}_\varepsilon = (\mathbf{A}_t^\top \mathbf{A}_t + \varepsilon \mathbf{I}_n)^{-1/2} = \mathbf{V}\mathbf{D}_\varepsilon \mathbf{V}^\top, \qquad \mathbf{D}_\varepsilon = \operatorname{diag}\Big((\sigma_j^2 + \varepsilon)^{-1/2}\Big)_{j=1}^{n}.$$

The score with the normalization of (2) is $U_z^{\mathrm{rp}} = \eta_t \langle \mathbf{G}_z \mathbf{P}_\varepsilon, \mathbf{R} \rangle_F$. Since

$$\mathbf{G}_z \mathbf{P}_\varepsilon = \mathbf{U}\mathbf{B}_z \mathbf{D}_\varepsilon \mathbf{V}^\top + \mathbf{U}_\perp \mathbf{C}_z \mathbf{D}_\varepsilon \mathbf{V}^\top,$$

taking the inner product with $\mathbf{R}$ and using the same coordinate uncorrelation assumption gives

$$\operatorname{Var}(U_z^{\mathrm{rp}}) = \eta_t^2 \left[\sum_{i,j} \tau_{ij}^2 \frac{[\mathbf{R}_\parallel]_{ij}^2}{\sigma_j^2 + \varepsilon} + \sum_{a,j} \nu_{aj}^2 \frac{[\mathbf{R}_\perp]_{aj}^2}{\sigma_j^2 + \varepsilon} \right]. \tag{94}$$

Without coordinate uncorrelation, the corresponding identity is $\operatorname{Var}(U_z^{\mathrm{rp}}) = \eta_t^2 \mathbf{p}^\top \mathbf{\Gamma} \mathbf{p}$, where $\mathbf{p} = \operatorname{vec}(\mathbf{R}\mathbf{P}_\varepsilon)$. Both scores are fixed linear functionals of $\operatorname{vec}(\mathbf{G}_z)$; if this vector is Gaussian, the scores are jointly Gaussian, possibly degenerate.

For any $\mathbf{G} = \mathbf{U}\mathbf{B}\mathbf{V}^\top + \mathbf{U}_\perp \mathbf{C}\mathbf{V}^\top$, subtracting the proxy from (88) gives

$$\begin{aligned} \mathcal{L}_t[\mathbf{G}] - \mathbf{G}\mathbf{P}_\varepsilon = \mathbf{U}\big(\mathbf{\Omega}(\mathbf{B}) - \mathbf{B}\mathbf{D}_\varepsilon\big)\mathbf{V}^\top \\ + \mathbf{U}_\perp \mathbf{C}\big(\mathbf{\Sigma}^{-1} - \mathbf{D}_\varepsilon\big)\mathbf{V}^\top. \end{aligned} \tag{95}$$

The two blocks are orthogonal, hence

$$\begin{aligned} \|\mathcal{L}_t[\mathbf{G}] - \mathbf{G}\mathbf{P}_\varepsilon\|_F^2 = \|\mathbf{\Omega}(\mathbf{B}) - \mathbf{B}\mathbf{D}_\varepsilon\|_F^2 \\ + \big\|\mathbf{C}(\mathbf{\Sigma}^{-1} - \mathbf{D}_\varepsilon)\big\|_F^2. \end{aligned} \tag{96}$$

At $\varepsilon = 0$, the proxy agrees with the derivative on the column-space complement, but generally not on the parallel block. For $\varepsilon > 0$, the complement weights also differ when that block is present. Since $\mathbf{P}_\varepsilon$ is invertible, the proxy annihilates no nonzero matrix, including radial perturbations. Their scalar alignments can nevertheless vanish for particular references. Thus radial invariance of the ideal polar map is not a radial-collapse guarantee for the deployed proxy.

Normalization and scope. The coefficient α is the candidate's weight in the local matrix perturbation, determined by the batch and momentum conventions; it is not universally $1/B$. The displayed proxy score omits this weight, as in (2). For equally weighted scores,

$$|U_z^{\mathrm{Muon}} - \alpha U_z^{\mathrm{rp}}| \leq \eta_t \alpha \, \|\mathbf{R}\|_F \, \|\mathcal{L}_t[\mathbf{G}_z] - \mathbf{G}_z \mathbf{P}_\varepsilon\|_F.$$

Multiplying any score by a fixed scalar multiplies its variance by the square of that scalar; any additional block-update rescaling must therefore be retained in calibrated comparisons. All variance identities above concern the specified linear scores, not the finite nonlinear alignment $\eta_t \Phi_z(\alpha)$. The Taylor estimate in (90) is local and must be controlled separately when relating their variances.

D.5 Optimal Candidate Multiplier (Lemma 1)

We provide the detailed proof of Lemma 1 here.

Proof. Following Theorem 1, the per-step selection gain is $G_t = c_q\,\sigma_U(t)$ with $q = 1/\lambda$. A step draws $\tilde{B} = \lambda B$ candidate rows and trains on B of them. Denote sequence length L, *OptiSelect streams and scores* λBL tokens while *training the model on* BL. Therefore, under a budget counted in tokens read, we quantify the selection gain per-read-token as

$$\frac{G_t}{\lambda BL} \;\propto\; \frac{c_q(\lambda)}{\lambda} \;=:\; f(\lambda), \qquad \lambda \in (1,\infty). \tag{97}$$

Let $U \sim \mathcal{N}(\mu,\sigma^2)$, define the density, CDF and upper tail functions for $z \sim \mathcal{N}(0,1)$ as

$$\phi(z) \;=\; \tfrac{1}{\sqrt{2\pi}}\,e^{-z^2/2}, \qquad \Phi(z) \;=\; \int_{-\infty}^{z}\phi(u)\,\mathrm{d}u, \qquad \bar{\Phi} \;=\; 1-\Phi,$$

with Φ^{-1} denoting the quantile function.

Let hard top-q selection retain the largest q-fraction, i.e. those U above the $(1-q)$-quantile $u_q = \mu + \sigma\,\Phi^{-1}(1-q)$. Writing $Z = (U-\mu)/\sigma \sim \mathcal{N}(0,1)$ and $a = \Phi^{-1}(1-q)$, the truncated-normal mean is

$$\mathbb{E}[Z \mid Z > a] \;=\; \frac{1}{\Pr[Z>a]}\int_a^{\infty} z\,\phi(z)\,\mathrm{d}z \;=\; \frac{[-\phi(z)]_a^{\infty}}{1-\Phi(a)} \;=\; \frac{\phi(a)}{1-\Phi(a)}, \tag{98}$$

using $\phi'(z) = -z\phi(z)$, so that $\int_a^{\infty} z\phi(z)\mathrm{d}z = \phi(a)$. By construction $1-\Phi(a) = q = \frac{1}{\lambda}$, hence selection lifts the mean utility by $\sigma\,\phi(a)/q$ and

$$c_q \;=\; \frac{\phi\big(\Phi^{-1}(1-q)\big)}{q}. \tag{99}$$

Therefore, plug in (97), we have:

$$f(\lambda) \;=\; \frac{c_q(\lambda)}{\lambda} \;=\; \phi\big(\Phi^{-1}(1-1/\lambda)\big). \tag{100}$$

Write $z(\lambda) = \Phi^{-1}(1-1/\lambda)$, so $\Phi(z) = 1-1/\lambda$. Differentiating implicitly,

$$\phi(z)\,z'(\lambda) \;=\; \frac{1}{\lambda^2} \qquad\Longrightarrow\qquad z'(\lambda) \;=\; \frac{1}{\lambda^2\,\phi(z)} \;>\; 0.$$

Then, plug in $\phi'(z) = -z\,\phi(z)$ with the chain rule, we have:

$$f'(\lambda) \;=\; \phi'(z)\,z'(\lambda) \;=\; \big(-z\,\phi(z)\big)\cdot\frac{1}{\lambda^2\phi(z)} \;=\; -\frac{z(\lambda)}{\lambda^2} \;=\; -\frac{\Phi^{-1}(1-1/\lambda)}{\lambda^2}. \tag{101}$$

We then find the optimum of (100) by introducing an involution $\lambda \mapsto \lambda/(\lambda-1)$, which maps $(1,\infty)$ onto itself and corresponds to $q \mapsto 1-q$. Because $\Phi^{-1}(q) = -\Phi^{-1}(1-q)$ and ϕ is even, (100) gives

$$f\left(\frac{\lambda}{\lambda-1}\right) \;=\; \phi\big(\Phi^{-1}(1/\lambda)\big) \;=\; \phi\big(-\Phi^{-1}(1-1/\lambda)\big) \;=\; f(\lambda),$$

so the efficiency curve is symmetric under that involution, i.e. $f(1.5) = f(3)$ and $f(1.25) = f(5)$ exactly. An involution of an interval has exactly one fixed point here, $\lambda = \lambda/(\lambda-1) \iff \lambda = 2$, and a strictly unimodal function invariant under it must attain its extremum there.

Therefore, the multiplier $\lambda^\star = 2$ sets the unique optimum.

□

E EXPERIMENT CONFIGURATIONS

E.1 MODEL ARCHITECTURE

Configurations. We follow Llama architecture with RMSNorm pre-normalization, SwiGLU and RoPE positional embedding for all the LLM pretraining runs in Section 5. We apply GPT2 BPE tokenizer with vocabulary size 50,304.

Table 3: Model architecture configurations.

Scale	d_{model}	layers	heads	seq. len.	tokens/step
124M	768	12	12	512	32,768
720M	2048	12	16	512	131,072

Infrastructure. All runs use single GH200 nodes (4 GPUs, DDP) on the Clariden cluster, inside an NGC PyTorch container. FineWeb-HQ and FinePhrase datasets are loaded in streaming.

E.2 TRAINING ARGUMENTS

On both 124M and 720M model runs, we apply cosine schedule decaying to $0.01 \times \eta_{\max}$ with 2,000 warmup steps and weight decay 0.1, the optimal learning rate are determined by a grid sweep per-optimizer and per-scale, according to Appendix E.4. The 124M models are trained with a global batch size $64 \times 512 = 32{,}768$ tokens across 4 GH200 GPUs, while the 720M models are trained with a global batch size $256 \times 512 = 131{,}072$ tokens On both standard training and *OptiSelect*, the models are train from scratch for 128k steps, consuming 4.2B tokens in total.

E.3 SELECTION PROXY DETAILS

The proxy set holds 4,096 densely packed 512-token sequences split evenly over ARC-Easy, ARC-Challenge, HellaSwag, PIQA, and SciQ, each document rendered as the task prompt concatenated with its *correct* completion. The proxy gradient $\mathbf{g}_{\mathcal{V},t}$ is recomputed every refresh interval (1000 steps) on the corresponding model checkpoint. The proxy refresh interval is determined by the ablation of Appendix E.5.

E.4 LEARNING RATE SWEEP

Switching a run to OptiSelect changes which data the optimizer sees, and could in principle move the optimizer's best step size with it. If it did, every OptiSelect run would need its own learning-rate search and the cost of using the method would roughly double. This appendix settles the question at 124M, and the answer is what licenses the cheaper protocol used at 720M.

LR Sweep on 124M runs On 124M models, we grid-search the learning rate over four geometrically spaced rates across 7 optimizers, on OptiSelect and standard training respectively.

According to Table 4 and Figure 8, we find **The optimal learning rate is transferrable on most optimizers.** All seven optimizers select the *same* learning rate under standard training and *OptiSelect* paradigm according to its downstream performance. Therefore, a run can adopt the optimal learning rate from its standard-training rate in the *OptiSelect* mode.

720M Learning-Rate Search Given the transferability of learning rate, we swept five optimizers in *standard* training mode at the full 720M configuration (16 GPUs, 256 sequences per step, 2k steps $=$ 262M tokens). We select the optimum across four grids according to Figure 9 and directly apply the same optimal learning rate on *OptiSelect*.

Table 4: **Downstream loss ("dwn") and FineWeb Validation loss ("val") across a sweep of learning rates at 124M.** Bold marks the optimal learning rate on the *downstream loss*, which are taken for experiments.

Optimizer	Mode	val / dwn at $\eta = \{0.25,\ 0.5,\ 1,\ 2\}\times$ default			
AdamW	std	3.692 / 3.963	3.610 / 3.865	3.579 / **3.840**	3.642 / 3.902
	sel	3.733 / 3.850	3.650 / 3.771	3.616 / **3.745**	3.949 / 4.080
AdEMAMix	std	3.680 / 3.950	3.617 / 3.885	3.599 / **3.863**	4.206 / 4.446
	sel	3.724 / 3.855	3.659 / 3.780	3.635 / **3.763**	4.484 / 4.646
Sophia	std	3.796 / 4.062	3.687 / 3.947	3.647 / **3.908**	3.730 / 3.979
	sel	3.828 / 3.950	3.724 / 3.856	3.679 / **3.804**	3.737 / 3.891
D-Muon	std	3.777 / 4.043	3.658 / 3.931	3.607 / **3.870**	3.622 / 3.877
	sel	3.789 / 3.971	3.675 / 3.851	3.620 / **3.794**	3.613 / 3.795
Lion	std	3.611 / 3.867	3.585 / **3.852**	3.603 / 3.862	5.697 / 5.946
	sel	3.638 / 3.841	3.609 / **3.834**	3.629 / 3.835	5.562 / 5.742
Signum	std	3.840 / 4.099	3.740 / 3.996	3.709 / **3.964**	3.867 / 4.107
	sel	3.858 / 4.059	3.761 / 3.967	3.727 / **3.934**	3.819 / 4.009
SGD	std	6.996 / 7.190	6.878 / 7.073	6.728 / 6.930	6.558 / **6.763**
	sel	6.992 / 7.160	6.885 / 7.054	6.722 / 6.905	6.506 / **6.648**

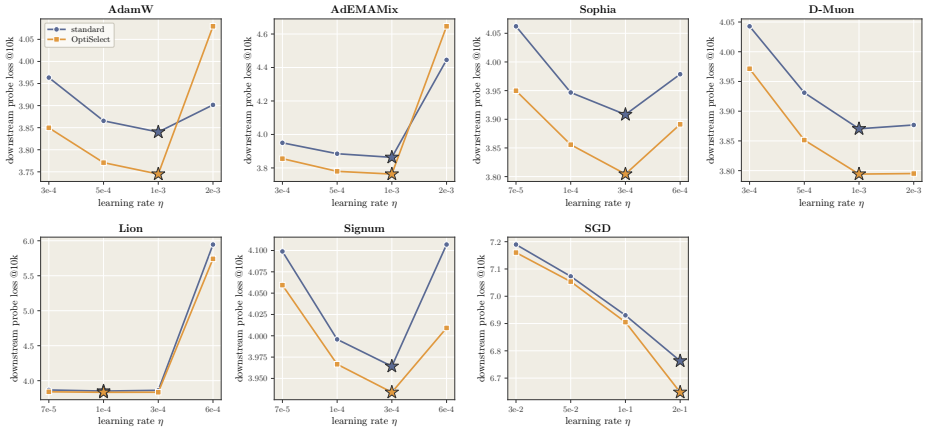


Figure 8: **Downstream loss in 124M learning-rate sweep.** Stars ⋆ mark the optimum for each optimizer.

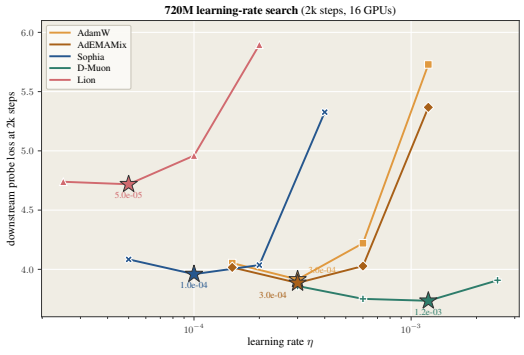


Figure 9: **Downstream loss in 720M learning-rate search.** Stars ⋆ mark the optimum for each optimizer.

E.5 Proxy Refresh Interval

Recomputing $\mathbf{g}_{\mathcal{V},t}$ costs a backward pass over the packed proxy set, so the engine caches it and reuses it for Δ steps. This trades a per-step cost against a staleness bias: between refreshes the reference gradient is evaluated at parameters the model has since left. We sweep $\Delta \in \{500, 1000, 5000, 10000\}$ on `AdamW` at its best rate ($\eta = 10^{-3}$, 10k steps, everything else held fixed) and report the results in Figure 10, Table 5. Since the cost is essentially free and the benefit saturates by $\Delta = 1000$ (3.7178 vs. 3.7165 nats at $\Delta = 500$), we set $\Delta = 1000$ as the default.

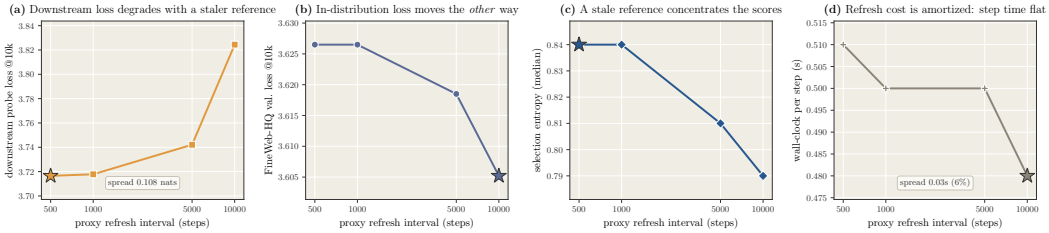


Figure 10: **Ablation on Proxy-refresh Frequency.** `AdamW` at $\eta = 10^{-3}$, 10k steps, varying only the refresh interval Δ; stars mark the best cell. (a) **The downstream loss** degrades monotonically proxy-fresh frequency; while (b) **Held-out FineWeb-HQ loss** moves in an *opposite* way. (c) **Selection entropy** falls as Δ grows, i.e. a stale reference makes the scores *look* more decisive while aligning them with a stale target. (d) **Step time is flat to within** 0.03 **s (**6%**)**: the refresh is amortized over Δ steps and never dominates.

Table 5: Proxy refresh sweep, `AdamW` at $\eta = 10^{-3}$, 10k steps at 124M. t_{step} is measured as median wall-clock per optimizer step.

refresh Δ (steps)	**500**	**1000**	5000	10000
downstream loss ↓	**3.7165**	**3.7178**	3.7421	3.8244
FineWeb-HQ val. loss	3.6265	3.6265	3.6185	**3.6052**
selection entropy	0.84	0.84	0.81	0.79
t_{step} (s)	0.51	0.50	0.50	0.48

F Extended Experimental Results

We present an extended analysis on the experimental results from section §5, at both 124M and 720M model scales.

F.1 Complete scores at 124M

Losses. We directly compare across various optimizers under standard training and *OptiSelect* in terms of average loss across all downstream tasks (Figure 11a,b) and validation loss on a held-out set from FineWeb-HQ (Figure 11c,d). The full loss values are presented in Table 6.

Figure 11 demonstrates that data selection improves downstream loss with every optimizers, while the ranking of optimizers is largely preserved at 124M, with `AdamW` and `AdEMAMix` takeing the leading position. In comparison, `Lion` and `D-Muon` falls behind other optimizers in rank despite its improvements on raw downstream performance. However, *OptiSelect* does not affect the optimizer ranking upon training distribution, i.e. the ranks upon held-out FineWeb-HQ loss are constant between standard training and *OptiSelect*.

Accuracy. Table 7 provides the complete accuracy scores across 9 in-domain and out-of-domain tasks. *OptiSelect* outperforms standard training with the equal amount of trained tokens in both in-domain and out-of-domain tasks, while falling behind with an equal amount of read tokens ($\lambda\times$ of trained tokens). Across optimizers, `AdEMAMix` achieves the best accuracy across in-domain tasks, while `AdamW` achieves the best score on out-of-domain ones. In comparison, `Lion` underperforms standard baseline in the average in-domain accuracy and `Sophia` underperforms in the average accuracies of out-of domain tasks.

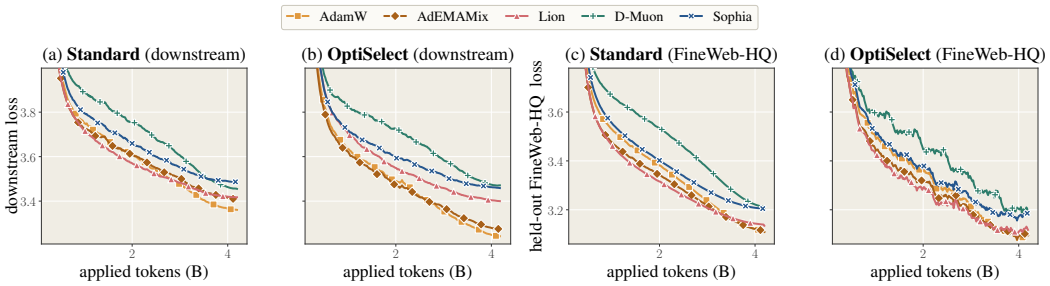


Figure 11: **Downstream loss and Held-out Fineweb-HQ loss compared across all optimizers at 720M.** (i) **Standard training**, (ii) ***OptiSelect***.

Table 6: **Downstream loss and Held-out Fineweb-HQ loss on 124M models.**

	Held-out FineWeb-HQ loss↓			Downstream loss↓		
Optimizer	std@128k	sel@128k	Δ	std@128k	sel@128k	Δ
`AdamW`	**3.107**	3.094	-0.013	**3.361**	**3.241**	-0.119
`AdEMAMix`	3.108	**3.093**	-0.014	3.410	3.274	-0.135
`Sophia`	3.198	3.179	-0.019	3.486	3.458	-0.028
`Lion`	3.133	3.116	-0.016	3.417	3.399	-0.018
`D-Muon`	3.204	3.195	-0.008	3.455	3.471	+0.015

Table 7: **124M zero-shot accuracy (%) per task.** *OptiSelect* outperforms standard training with equal amount of trained tokens, while falls behind with the equal amount of read tokens ($\lambda\times$ of trained tokens). Bold marks the best performance scores. Red color marks the slots where *OptiSelect* underperformed.

		In-distribution						Out-of-distribution				
Optimizer	**Checkpoint**	PIQA	HSwag	ARC-e	ARC-c	SciQ	**mean**	Wino	WSC	CSQA	OBQA	**mean**
AdamW	std@128k	64.80	31.83	44.11	23.38	65.40	45.90	51.14	36.54	19.66	28.80	34.03
	sel@128k	64.80	32.56	44.32	25.51	65.20	**46.48**	52.96	41.35	19.74	29.00	**35.76**
	sel@64k	62.08	30.56	42.93	23.72	62.40	44.34	50.91	36.54	19.49	28.00	33.73
AdEMAMix	std@128k	63.71	31.98	43.01	25.51	65.40	45.92	50.83	36.54	19.57	29.40	34.09
	sel@128k	65.18	32.50	43.81	25.26	66.80	**46.71**	54.14	36.54	20.64	26.60	**34.48**
	sel@64k	62.51	29.91	41.88	23.46	61.80	43.91	51.38	36.54	19.57	29.00	34.12
Sophia	std@128k	62.95	31.27	41.71	24.32	58.80	43.81	49.64	38.46	19.66	27.80	**33.89**
	sel@128k	62.40	31.26	43.22	24.15	62.80	**44.77**	50.28	36.54	19.57	26.40	33.20
	sel@64k	62.35	29.58	41.12	25.34	60.20	43.72	50.59	36.54	19.57	28.00	33.68
Lion	std@128k	63.44	32.39	43.18	23.21	65.00	**45.44**	50.75	36.54	19.49	27.60	33.60
	sel@128k	63.71	32.10	42.47	25.51	62.20	45.20	51.46	37.50	19.57	28.00	**34.13**
	sel@64k	61.86	30.17	40.99	24.23	59.70	43.39	52.41	36.54	19.57	27.60	34.03
D-Muon	std@128k	64.58	32.15	41.96	24.15	62.60	45.09	52.88	36.54	19.74	25.00	33.54
	sel@128k	63.00	32.62	42.47	24.06	64.20	**45.27**	50.99	40.38	19.66	28.00	**34.76**
	sel@64k*	61.15	29.84	40.19	23.63	60.00	42.97	49.57	38.46	19.57	27.20	33.70

F.2 COMPLETE SCORES AT 720M

Losses. According to Figure 12 and Table 8. At 720M, `D-Muon` ranks at the first under standard training while falling behind of `AdamW` and `AdEMAMix` with *OptiSelect* in downstream losses. Whereas for held-out FineWeb-HQ loss, `Muon` keeps its leading position while `AdamW`, `AdEMAMix` and `Lion` demonstrate a degradation compared with standard training.

Accuracy. Table 9 provides the complete accuracy scores across 9 in-domain and out-of-domain tasks at 720M. All the optimizers outperform standard training in in-domain accuracies, while `Muon` and `AdEMAMix` underperform in average out-of-domain accuracy.

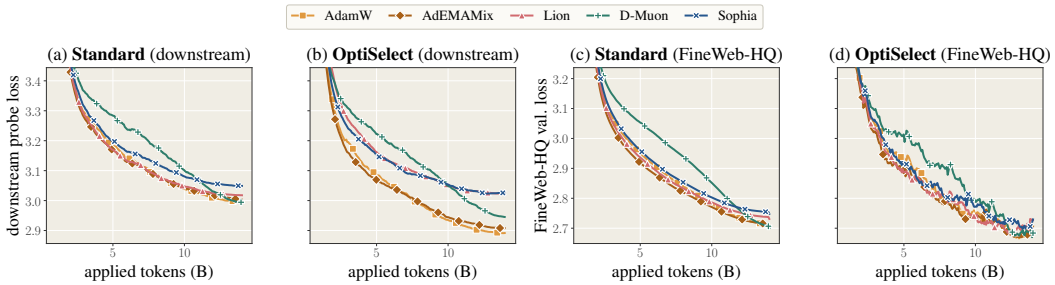


Figure 12: **Downstream loss and Held-out Fineweb-HQ loss compared across all optimizers at 720M.** (i) **Standard training**, (ii) ***OptiSelect***.

Table 8: **Downstream loss and Held-out Fineweb-HQ loss on 720M models.**

	Held-out FineWeb-HQ			Downstream probe		
Optimizer	std@107k	sel@107k	Δ	std@107k	sel@107k	Δ
`AdamW`	2.704	2.714	+0.010	2.995	2.891	-0.104
`AdEMAMix`	2.705	2.710	+0.005	3.004	**2.908**	-0.096
`Sophia`	2.745	2.736	-0.009	3.049	3.026	-0.023
`Lion`	2.729	2.731	+0.002	3.018	3.022	+0.004
`D-Muon`	2.696	**2.684**	-0.012	2.993	2.944	-0.049

Table 9: **720M zero-shot accuracy (%), per task.**

		In-distribution						Out-of-distribution				
Optimizer	**Checkpoint**	PIQA	HSwag	ARC-e	ARC-c	SciQ	**mean**	Wino	WSC	CSQA	OBQA	**mean**
`AdamW`	std@107k	69.45	46.03	53.89	28.37	75.85	54.72	50.87	45.19	19.49	32.50	**37.01**
	sel@107k	70.51	47.05	56.44	28.75	76.30	**55.81**	52.88	36.54	20.72	33.00	35.79
	sel@53.5k	68.72	41.35	52.31	27.39	73.00	52.55	49.96	36.54	19.74	30.20	34.11
`AdEMAMix`	std@107k	69.29	46.04	54.19	27.60	74.50	54.32	52.49	42.31	19.57	32.60	36.74
	sel@107k	70.13	47.07	55.43	27.82	76.00	**55.29**	52.80	54.81	18.59	34.00	40.05
	sel@53.5k	68.61	42.27	51.52	27.13	73.40	52.59	51.22	59.62	20.07	32.40	**40.83**
`Sophia`	std@107k	69.04	44.14	52.21	26.96	73.65	53.20	50.91	37.50	21.09	32.90	35.60
	sel@107k	69.59	45.41	53.41	28.58	75.00	**54.40**	50.99	47.12	21.29	32.20	**37.90**
	sel@53.5k	67.68	41.66	52.48	27.13	71.20	52.03	51.38	36.54	20.88	31.40	35.05
`Lion`	std@107k	69.94	44.95	53.85	27.69	72.30	53.75	50.59	36.54	20.84	30.70	34.67
	sel@107k	71.11	45.38	52.90	27.30	74.40	**54.22**	51.93	42.31	20.23	32.60	**36.77**
	sel@53.5k	67.85	41.62	48.57	26.54	70.80	51.07	50.67	38.46	20.31	30.40	34.96
`D-Muon`	std@107k	70.67	47.49	54.84	28.24	73.55	54.96	52.92	58.65	20.43	32.50	**41.13**
	sel@107k	70.57	49.91	55.51	29.52	76.60	**56.42**	54.06	45.19	20.48	33.60	38.33
	sel@53.5k	67.57	41.09	47.64	26.79	70.30	50.68	52.96	36.54	19.25	31.00	34.94

G Data Selection under Rephrasing

G.1 Details on the Downstream Tasks

We categorize all five downstream tasks into two groups according their probing formats: (1) ***QA-formatted***: A task is *QA-formatted* if the prompt poses an explicit question and terminates with an answer slot, so the model is scored on the Question-Answering ability; (2) ***completion-formatted*** A task is is *completion-formatted* if it feeds a plain textual stem to the model, and every option is a continuation of it. The model is scored on its ability on continuation.

According to the evaluation harness on each task, we categorize ARC-Easy, ARC-Challenge, PIQA, SciQ, CommonSense QA, Openbook QA as **QA-formatted**; and HellaSwag, WinoGrande, WSC are **completion-formatted**.

G.2 Extended Results on Rephrasing

We present the complete results on the *OptiSelect* $\times$ Format rephrasing experiments in §5.3. According to Table 10, *OptiSelect* consistently outperforms standard training across various rephrasing mixtures in terms of average accuracy on in-domain proxy tasks and out-of-domain benchmarks. Notably, both standard training and *OptiSelect* achieve the best performance across all 9 tasks at 50% rephrasing rates, which mixes original FineWeb and FinePhrase documents by $1:1$ ratio.

Table 10: **Format rephrasing $\times$ *OptiSelect*, zero-shot accuracy (%).**

		In-Domain						Out-of-Domain					
Rephrased	**Mode**	PIQA	HSwag	ARC-e	ARC-c	SciQ	**mean**	Wino	WSC	CSQA	OBQA	**mean**	overall-mean
0%	standard	**64.74**	29.59	47.81	**20.65**	**75.10**	47.58	51.14	36.54	19.66	16.60	30.98	40.20
	OptiSelect	**64.74**	**30.01**	**50.04**	20.31	74.10	**47.84**	**52.01**	**44.23**	**19.74**	**17.40**	**33.35**	**41.40**
50%	standard	64.47	29.87	51.94	23.04	75.90	49.04	50.28	38.46	**19.57**	**18.00**	31.58	41.28
	OptiSelect	**64.91**	**30.13**	**53.24**	**23.29**	**77.00**	**49.72**	**52.17**	**40.38**	**19.57**	17.60	**32.43**	**42.03**
100%	standard	63.71	30.05	54.00	**23.21**	75.60	49.31	49.96	**36.54**	19.49	18.40	31.10	41.22
	OptiSelect	**64.91**	**30.23**	**54.38**	22.70	**77.70**	**49.98**	**52.17**	**36.54**	**19.74**	**19.40**	**31.96**	**41.97**