

HOW INNER OPTIMIZERS SHAPE REPLICA SCALING

Eugene Belilovsky^{1,2} Stefan Horoi^{1,3} Edouard Oyallon⁴ Benjamin Thérien^{1,3}

¹Mila ²Concordia University ³Université de Montréal ⁴CNRS

ABSTRACT

Training large language models (LLMs) across geographically distributed and poorly interconnected compute pools or data centers has become a topic of significant interest. This has led to increased study of local optimization methods such as DiLoco for LLMs, especially those that use local (or inner) adaptive or preconditioned optimizers, which are critical for high-performance LLM training. However, a central challenge remains: it is well known that increasing the number of model replicas (replica scaling) while fixing the total training budget leads to monotonically decreasing performance. Existing work has empirically demonstrated that the choice of inner optimizer (Muon v.s. AdamW) can significantly improve this trend, but offers only limited explanation of the performance difference. In this work, we consider a range of inner-optimizers and identify two important factors that impact replica scaling: (1) the covariance structure of the pseudo-gradients and (2) the impact of batch size on the pseudogradient’s noise scale. Based on these insights, we theoretically and empirically demonstrate that whitening-based preconditioning is a powerful inner-optimizer desideratum. Our empirical evaluation extends the results to reveal that whitening-based inner-optimizers achieve strong replica scaling (for 8 – 96 replicas), with performance relative to DP improving as parameters and data are scaled.

1 INTRODUCTION

The computational scale of language-model training increasingly exceeds the capacity of a single site. Training across geographically separated clusters can pool otherwise fragmented resources, but the connectivity between these clusters is poor, leading to significant communication bottlenecks. Local optimization methods address this constraint by allowing replicas to optimize independently for several steps between parameter or pseudo-gradient averages (Douillard et al., 2023; 2025; Liu et al., 2024) often drastically reducing the frequency of communication.

A central difficulty in practical local optimization is replica scaling: at a fixed token budget, increasing the number of replicas often degrades the final model performance (Charles et al., 2025) relative to its non-local data-parallel (DP) counterpart. This can limit the broad applicability of these methods. It has been shown that the replica scaling can improve with increasing model size (Charles et al., 2025). Recent empirical results show that modifying either the inner optimizer (Thérien et al., 2025) or the outer aggregation rule (Horoi et al., 2026) can further improve the scalability and efficiency of distributed training. However, the impact of inner optimizers on scaling is not well understood. More broadly, the primary sources of divergence from standard non-local training are not well understood.

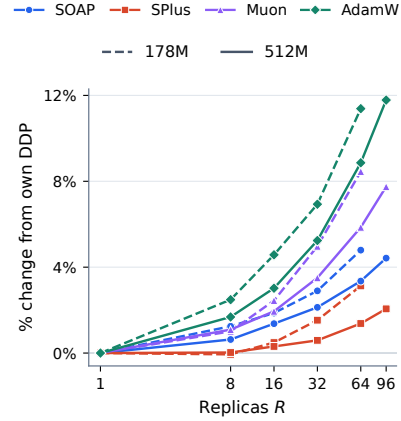


Figure 1: Local optimization performance as replicas are scaled. Each local optimizer is compared to its own DP baseline, to eliminate confounding replica scaling with absolute loss. Optimizers that whiten the pseudogradient covariance achieve the strongest worker scaling. SPlus is the best proxy for whitening. We note that SOAP’s stronger absolute performance sometimes leads to the best results despite its poorer worker scaling. Notably, we also observe a strong scaling trend relative to DP for all inner-optimizers.

Classical analyses of local SGD and federated averaging study the discrepancy accumulated over multiple local steps, often with heterogeneous client data. The primary source of performance degradation is often attributed to client drift or other notions of divergence of model replicas along their trajectory (Karimireddy et al., 2020). Much of this work and theory (Stich, 2019; Karimireddy et al., 2020; Yu et al., 2019) focuses on SGD and momentum SGD as the inner optimizer, a single local step from common parameters and optimizer states that commute with averaging: averaging the updates gives the same result as updating with the averaged gradient. Nonlinear adaptive inner optimizers need not satisfy this property and are much less well understood in the context of this literature. Notably, a discrepancy can arise on the first step, even with identical starting states and identically distributed data, because transforming each local gradient before averaging differs from transforming the averaged gradient. Understanding replica scaling therefore requires examining how the inner optimizer transforms gradient noise, in addition to how parameters drift over successive local steps.

In this work, we focus on a broad range of popular *adaptive* or *preconditioner* based optimizers and identify two critical properties that can greatly impact scaling: (a) the covariance structure of the pseudo-gradient induced by the local optimization (b) how the optimizers deal with statistical estimation of second moments in the low batch size setting.

Our contributions can be summarized as follows:

- We demonstrate, both empirically and theoretically, the importance of the pseudogradient covariance structure induced by the choice of inner optimizer. Specifically, we link the choice of inner optimizer to the maximum discrepancy with respect to the DP update direction.
- We identify the importance of appropriately scaling β_2 when using second-moment optimizer states, validating our findings through theoretical analysis and experiments. This reveals that the recently introduced IsoLoCo method (Horoi et al., 2026), using an Adam inner optimizer, can outperform the much more computationally expensive MuLoCo (Thérien et al., 2025).
- Our results demonstrate that using a SOAP (Vyas et al., 2025) or SPlus (Frans et al., 2025) inner optimizer can lead to very effective replica scaling, obtaining significantly improved performance relative to existing methods from the literature (as much as 6% less degradation relative to DP at 96 replicas).

2 RELATED WORK

Local-step methods and DiLoCo. Local SGD (Stich, 2019) and federated averaging (McMahan et al., 2017) reduce communication by taking several local steps between averaging events. DiLoCo (Douillard et al., 2023) adapts this to LLM pretraining by utilizing a large number of inner steps, AdamW (Loshchilov & Hutter, 2019) as the inner optimizer, and SGD with Nesterov momentum on the aggregated pseudo-gradients as the outer optimizer. Follow-ups relax synchrony (Liu et al., 2024) and overlap or shard the exchange of information between replicas (Douillard et al., 2025). Charles et al. (2025) establish scaling laws for DiLoCo and document the central difficulty we study: adding replicas at a fixed token budget degrades quality, with the optimal hyperparameters shifting as R grows. Our analysis complements these scaling laws by tracing the degradation back to local state estimation and to the nonlinear optimizer transforms applied before aggregation. Local methods can also be complemented with local optimization by using Sparse parameter averaging (SPARTA; Reed & Baoumy, 2025) or Top- k sparsification SparseLoCo (Sarfi et al., 2025). IsoLoco Horoi et al. (2026) views DiLoCo’s outer step as task-arithmetic model merging and replaces plain pseudo-gradient averaging with Iso-C merging (Marczak et al., 2025). For each weight matrix, it takes the SVD of the averaged pseudo-gradient and replaces all singular values with their mean, giving $\bar{\sigma}UV^\top$. It then applies the usual Nesterov momentum outer step. This was shown to improve worker scaling over DiLoCo (Horoi et al., 2026).

Small-batch training and adaptive-optimizer scaling rules. Scaling rules relate optimizer hyperparameters to batch size (Malladi et al., 2022); most directly, Marek et al. (2025) show that Adam trains stably at very small batches provided β_2 is set so that the second moment’s half-life is fixed in *tokens* rather than in steps. Replica scaling in DiLoCo induces exactly this small-batch regime for

each individual replica. Section 4.4 transfers the β_2 scaling rule to replica scaling; to our knowledge this hasn't been studied has not been studied in the local-step setting previously.

Spectral and second-order optimizers. Shampoo (Gupta et al., 2018) preconditions with Kronecker-factored curvature estimates; SOAP (Vyas et al., 2025) runs Adam in Shampoo's eigenbasis; Muon (Jordan et al., 2024) orthogonalizes matrix updates with no accumulated second-moment statistics. SPlus (Frans et al., 2025) retains Shampoo's historical eigenbasis but signs momentum in that basis, avoiding cached eigenvalue magnitudes and SOAP's additional coordinate-wise second moment. SPlus uses this instant-sign transformation independently in local DiLoCo replica bases; our focus is its distributed covariance and replica-scaling behavior. Controlled single-stream comparisons find the ranking horizon-dependent (Wen et al., 2025), while a recent multi-billion-parameter study reports a slight edge for a stabilized SOAP over Muon at very large batch, at the price of memory and numerical-stability engineering (Khona et al., 2026). In the DiLoCo setting, MuLoCo (Thérien et al., 2025) uses Muon as the inner optimizer and reports more directionally consistent pseudo-gradients as R grows, while IsoLoCo (Horoi et al., 2026) treats outer aggregation as model merging and isotropizes the aggregated pseudo-gradient. Section 4 develops a complementary criterion: whether the stochastic update geometry emitted by each replica remains well conditioned under averaging.

3 BACKGROUND: THE DiLoCo FRAMEWORK

The methods in this paper can be seen as an instance of Algorithm 1 which is a generalization of local optimization from Reddi et al. (2021); Douillard et al. (2023) that utilizes both inner (on the local gradients) and outer optimization (on the pseudo-gradients). R replicas start each round from the shared parameters, each runs H steps of an inner optimizer on its own data stream. The resulting displacements Δ_r are uniformly averaged, and an outer optimizer applies the aggregate $\bar{\Delta}$ to the shared parameters from the start of the round. DiLoCo (Douillard et al., 2023) instantiates the inner optimizer as AdamW and the outer as Nesterov momentum. For example MuLoCo (Thérien et al., 2025) replaces the inner optimizer with Muon. IsoLoCo (Horoi et al., 2026) isotropizes $\bar{\Delta}$ before the outer step. The inner optimizer states s_r are never communicated: each is estimated locally from its replica's data stream.

Algorithm 1 The DiLoCo framework. All methods studied here are instances of this algorithm.

Require: initial parameters $\theta^{(0)}$; replicas $r \in \{1, \dots, R\}$; inner steps H ; inner optimizer INNER with per-replica state s_r ; outer optimizer OUTER with state S ; global batch size B ;

```

1: for outer round  $t = 0, \dots, T - 1$  do
2:   for all replicas  $r$  in parallel do
3:      $\theta_r \leftarrow \theta^{(t)}$ 
4:     for  $h = 1, \dots, H$  do
5:        $g \leftarrow \nabla_{\theta} \mathcal{L}(\theta_r; x_{r,h})$  ▷ local minibatch:  $B/R$  tokens
6:        $(\theta_r, s_r) \leftarrow \text{INNER}(\theta_r, g, s_r)$  ▷  $s_r$  is never synchronized
7:     end for
8:      $\Delta_r \leftarrow \theta^{(t)} - \theta_r$ 
9:   end for
10:   $\bar{\Delta} \leftarrow \frac{1}{R} \sum_r \Delta_r$  ▷ uniform average
11:   $(\theta^{(t+1)}, S) \leftarrow \text{OUTER}(\theta^{(t)}, \bar{\Delta}, S)$  ▷ Nesterov momentum in DiLoCo
12: end for
```

4 INNER OPTIMIZERS IMPACT ON REPLICA SCALING

The transformations applied to local updates by local optimizers can have a significant impact on the final aggregated pseudo-gradient. We investigate how the choice of inner optimizer shapes the distribution of the resulting pseudo-gradients and their deviation from the corresponding non-local updates. Section 4.1 introduces the optimizers considered in this work, Section 4.2 separates the update error (relative to non-local steps with the same base optimizer) into a **mean mismatch** term and a **stochastic error** term. It shows how whitening controls the largest directional variance, and how matching covariance structures across replicas makes uniform averaging optimal. Section 4.4 specializes this framework to one step from common state: under the stated Gaussian model, SPlus

admits a scalar calibration that matches the DP mean to first order in the global proposal’s signal-to-noise ratio. Finally, Section 4.4 bounds the Adam mean mismatch caused by fresh-gradient second moments and explains how appropriately adjusting β_2 limits that bound as local batches shrink. It then isolates a complementary advantage of SPlus: retaining the estimated basis without relying on coordinatewise scale estimates to control the covariance term.

4.1 BACKGROUND ON POPULAR OPTIMIZERS

In this section we present a number of popular optimizers using a unified notation to facilitate their common analysis and comparison. In particular, we include recent matrix-based optimizers such as Muon and those in the Shampoo family. The optimizers differ in how they retain or remove the magnitudes of a momentum proposal before averaging across replicas. We write these transformations explicitly so their mean and covariance can be studied in the same framework.

Consider a matrix parameter with stochastic gradient $G \in \mathbb{R}^{m \times n}$. An optimizer returns a direction U and an updated state after observing G ; its parameter displacement is $\Delta = \eta \text{vec}(U)$. We use subscripts $t-1$ and t for the incoming and updated optimizer states, respectively, and suppress the step index on the gradient and returned direction. The first-moment proposal is

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1)G. \quad (1)$$

We compare the following matrix-update cores, using standard EMA momentum to isolate their normalizations. For simplicity we omit standard deterministic bias corrections and weight decay from the presentation. The matrix-size scale γ of Muon is shown explicitly. All decay coefficients lie in $[0, 1)$; numerical damping is denoted by $\epsilon > 0$.

Adam and Shampoo. Adam uses an entrywise second moment in parameter coordinates (Kingma & Ba, 2015; Loshchilov & Hutter, 2019):

$$V_{A,t} = \beta_2 V_{A,t-1} + (1 - \beta_2)G^{\odot 2}, \quad U_{\text{Adam}} = M_t \oslash (\sqrt{V_{A,t}} + \epsilon). \quad (2)$$

Where U is the update. Here, $\odot$ and $\oslash$ denote elementwise multiplication and division, respectively, with $G^{\odot 2}$ representing the elementwise square. Shampoo instead maintains left and right Gram factors (Gupta et al., 2018):

$$\begin{aligned} L_t &= \beta_3 L_{t-1} + (1 - \beta_3)GG^\top, & R_t &= \beta_3 R_{t-1} + (1 - \beta_3)G^\top G, \\ U_{\text{Shampoo}} &= (L_t + \epsilon I)^{-1/4} M_t (R_t + \epsilon I)^{-1/4}. \end{aligned} \quad (3)$$

The above formulation is the canonical version of Shampoo. We note that some variants consider half-power instead of quarter (Vyas et al., 2025). We also note variants with grafting or additional normalizations are often used, which can impact the interpretation of the underlying preconditioner (Lin et al., 2026).

SOAP and SPlus. Let Q_L, Q_R be orthogonal eigenvector matrices obtained from the historical Gram factors above. Their action is the matrix rotation

$$\tilde{G} = Q_L^\top G Q_R, \quad \tilde{M}_t = Q_L^\top M_t Q_R. \quad (4)$$

SOAP (ShampoO with Adam in the Preconditioner’s eigenbasis) performs Adam-style normalization in this product basis (Vyas et al., 2025):

$$\begin{aligned} V_{S,t} &= \beta_2 V_{S,t-1} + (1 - \beta_2)\tilde{G}^{\odot 2}, \\ U_{\text{SOAP}} &= Q_L [\tilde{M}_t \oslash (\sqrt{V_{S,t}} + \epsilon)] Q_R^\top. \end{aligned} \quad (5)$$

In this work, we consider a signed update in a Shampoo basis, a variant of the recently introduced SPlus (Frans et al., 2025) optimizer, which does not utilize iterate averaging. This optimizer retains the momentum and historical basis but signs each projected entry:

$$U_{\text{SPlus}} = Q_L \text{sign}(\tilde{M}_t) Q_R^\top, \quad \text{sign}(0) = 0. \quad (6)$$

This approach can be seen as a simplified version of SOAP that removes the dependence on the second moment. In the following sections, we will further illustrate how this optimizer naturally arises in the context of local optimization with replica scaling. SOAP’s rotated normalization and SPlus sign transform have a whitening interpretation (Vyas et al., 2025; Frans et al., 2025; Lin et al., 2026). Section 4.2 gives the corresponding covariance identities under explicit assumptions and explains their impact on pseudo-gradient aggregation.

Muon. For a singular value decomposition $\mathbf{M}_t = \mathbf{U}_M \mathbf{\Sigma}_M \mathbf{V}_M^\top$ on the nonzero singular subspace, ideal Muon returns

$$\mathbf{U}_{\text{Muon}} = \gamma \text{polar}(\mathbf{M}_t) = \gamma \mathbf{U}_M \mathbf{V}_M^\top. \quad (7)$$

It replaces nonzero singular values by a common value (Jordan et al., 2024). In matched singular-vector coordinates, $\mathbf{U}_M^\top \mathbf{M}_t \mathbf{V}_M = \mathbf{\Sigma}_M$, so at its core Muon also has the form “rotate, remove magnitudes, rotate back.” Unlike SPlus, its basis is determined by the current proposal, not by historical covariance statistics. In practice, Newton-Schulz is used to approximate the polar map.

4.2 IMPACT OF COVARIANCE STRUCTURE ON LOCAL OPTIMIZATION

We consider the general local optimization setting (under any local optimizer) and separate the error relative to the expected DP update with the same optimizer into two terms: (1) **mean mismatch** and (2) **stochastic error**. The covariance analysis in this subsection addresses the stochastic term; Section 4.4 then addresses the mean mismatch. Following the pseudo-gradient interpretation of client displacements in Reddi et al. (2021) and Wang et al. (2024), we treat each replica’s complete local update as a random vector. All expectations and covariances below are conditional on the start-of-round parameters and the incoming optimizer states. Let $\mathbf{\Delta}_r \in \mathbb{R}^d$ be the complete H -step displacement returned by replica r . We decompose each local displacement into its conditional mean and a zero-mean stochastic component:

$$\mathbf{\Delta}_r = \boldsymbol{\mu}_r + \boldsymbol{\varepsilon}_r, \quad \boldsymbol{\mu}_r = \mathbb{E}[\mathbf{\Delta}_r], \quad \mathbb{E}[\boldsymbol{\varepsilon}_r] = 0, \quad \mathbf{C}_r = \text{Cov}(\mathbf{\Delta}_r). \quad (8)$$

Assume finite second moments, $\mathbb{E}[\|\mathbf{\Delta}_r\|_2^2] < \infty$ for all r , and zero cross-replica noise covariances, $\text{Cov}(\boldsymbol{\varepsilon}_r, \boldsymbol{\varepsilon}_s) = \mathbf{0}$ for all $r \neq s$. Independent local data streams, conditional on the start-of-round parameters and optimizer states, are sufficient to ensure zero cross-replica noise covariances. Set $\bar{\mathbf{\Delta}} = R^{-1} \sum_r \mathbf{\Delta}_r$ and $\bar{\boldsymbol{\mu}} = R^{-1} \sum_r \boldsymbol{\mu}_r$. Note that for the typical setting studied in Douillard et al. (2023); Thérien et al. (2025) with i.i.d. data and identical incoming optimizer states, $\boldsymbol{\mu}_r$ is the same for all replicas.

We now consider $\mathbf{d}_H^{\text{DP}}$: the displacement of the *same inner optimizer* run for H *synchronized* steps and same global batch size, in expectation. We allow a deterministic scalar calibration factor α_R , and define

$$\mathbf{b}_R = \alpha_R \bar{\boldsymbol{\mu}} - \mathbf{d}_H^{\text{DP}}, \quad \mathbf{C}_{\text{agg}} = \text{Cov}(\bar{\mathbf{\Delta}}) = \frac{1}{R^2} \sum_r \mathbf{C}_r. \quad (9)$$

For $H = 1$, α_R can represent a step-size change, for example. We now consider the following decomposition of the error in any direction and the corresponding bound:

Proposition 1 (Directional error decomposition). *The conditional bias–variance decomposition gives, for every unit vector $\mathbf{q}$,*

$$\underbrace{\mathbb{E}[(\mathbf{q}^\top (\alpha_R \bar{\mathbf{\Delta}} - \mathbf{d}_H^{\text{DP}}))^2]}_{\text{error relative to the DP mean}} = \underbrace{(\mathbf{q}^\top \mathbf{b}_R)^2}_{\text{mean mismatch}} + \underbrace{\alpha_R^2 \mathbf{q}^\top \mathbf{C}_{\text{agg}} \mathbf{q}}_{\text{stochastic error}}. \quad (10)$$

Consequently, the largest directional error is

$$\lambda_{\max}(\mathbf{b}_R \mathbf{b}_R^\top + \alpha_R^2 \mathbf{C}_{\text{agg}}) \leq \|\mathbf{b}_R\|_2^2 + \alpha_R^2 \lambda_{\max}(\mathbf{C}_{\text{agg}}). \quad (11)$$

Proof. Subtract and add $\alpha_R \bar{\boldsymbol{\mu}}$. Using the Rayleigh quotient gives (11). $\square$

The second term allows us to analyze the effect of inner optimizers that induce whitened updates. As we show below, whitening helps control the stochastic error by bounding the largest directional variance of the aggregated pseudo-gradient, while also promoting covariance matching across replicas. The mean mismatch term is more challenging to analyze; however in Section 4.4 we derive explicit expressions and bounds for several inner optimizers under a single-local-step setting. Moreover we will show in the empirical results that this term is generally dominated in the largest noise direction.

Whitening and directional variance. A linear map P whitens a covariance Σ if $P\Sigma P^\top = I$ (Kessy et al., 2018); a prescribed scalar multiple gives a spherical covariance at another scale. For any covariance C of dimension d and trace τ (fixed total variance), the following inequality holds:

$$\lambda_{\max}(C) \geq \tau/d, \quad \text{with equality precisely when } C = (\tau/d)I. \quad (12)$$

Thus, isotropy minimizes the *largest* directional variance at fixed total variance, precisely the covariance quantity appearing in (11). Moreover, a bound on the maximum directional variance of each replica carries over to the aggregate:

$$\lambda_{\max}(C_{\text{agg}}) \leq \frac{1}{R^2} \sum_r \lambda_{\max}(C_r).$$

Consequently, whitening local pseudo-gradients toward a common isotropic covariance controls the stochastic error.

Whitening and optimality of uniform averaging. Local update whitening has an additional, pre-aggregation benefit when inner optimizer trajectories produce different update distributions. For the stochastic term alone, a weighted aggregate has directional variance

$$\sum_{r=1}^R w_r^2 \mathbf{q}^\top C_r \mathbf{q}, \quad \sum_r w_r = 1.$$

Uniform weights, which are used by default in standard DiLoCo settings, minimize this quantity when the replicas have equal directional variances (Aitken, 1935); they do so in every direction when $C_1 = \dots = C_R$. Whitening each replica toward the same spherical target νI promotes precisely this agreement, since $\|C_i - C_j\| \leq \|C_i - \nu I\| + \|C_j - \nu I\|$.

Thus, local whitening both controls the worst-direction aggregate noise and makes the default uniform averaging well suited to the individual replica noise geometries. However, it does not control the mean mismatch $\mathbf{b}_R$, which is analyzed separately in Section 4.4.

4.3 IMPACT OF OPTIMIZER TRANSFORMATIONS ON UPDATE COVARIANCE

In this subsection, we analyze how the transformations induced by different inner optimizers affect the covariance structure of their updates under idealized assumptions, with particular emphasis on their whitening properties. This comparison is motivated by the eigenbasis interpretation of SOAP (Vyas et al., 2025), the whitening motivation of Frans et al. (2025), and the covariance-estimation analysis of Lin et al. (2026).

For this comparison, consider a random matrix proposal $M = M_t \in \mathbb{R}^{m \times n}$, with $\Sigma = \text{Cov}(\text{vec}(M)) \succ 0$. Let U denote the resulting update matrix, and define $d = mn$ and $C = \text{Cov}(\text{vec}(U))$. We treat optimizer bases and linear preconditioners as fixed independently of the sampled proposal, and define ideal normalization using the proposal’s *centered* covariance. Write

$$Q = Q_R \otimes Q_L, \quad \text{vec}(Q_L^\top M Q_R) = Q^\top \text{vec}(M).$$

We now examine how each optimizer transforms the covariance of the stochastic proposal and under which conditions the resulting update covariance is isotropic.

Adam. Suppose each entry of M is divided by its true standard deviation. Writing $D = \text{diag}(\Sigma_{11}, \dots, \Sigma_{dd})$, this gives

$$\text{vec}(U) = D^{-1/2} \text{vec}(M), \quad C = D^{-1/2} \Sigma D^{-1/2}.$$

Every diagonal entry of C is one, but $C_{ij} = \Sigma_{ij} / \sqrt{\Sigma_{ii} \Sigma_{jj}}$. Thus, even with access to the true coordinate variances, Adam’s normalization preserves the correlations present in M .

Shampoo. Consider the idealized case where the factors equal the population Gram matrices of a zero-mean input: $L = \mathbb{E}[MM^\top]$ and $R = \mathbb{E}[M^\top M]$. Assume the covariance is separable, $\Sigma = (R \otimes L) / \text{tr}(L)$, with positive-definite factors. The canonical Shampoo transformation then gives:

$$U = L^{-1/4} M R^{-1/4}, \quad C = \frac{R^{1/2} \otimes L^{1/2}}{\text{tr}(L)}.$$

The condition number of the resulting covariance, i.e., the ratio of its largest to smallest eigenvalues, is the square root of that of the input covariance. Thus, canonical Shampoo does not generally whiten the updates, as also discussed in Lin et al. (2026).

SOAP. Suppose the fixed rotation $\widetilde{M}_t = Q_L^\top M_t Q_R$ gives uncorrelated coordinates, and let $V_{S,t}$ denote their true variances: $(V_{S,t})_{ij} = \text{Var}((\widetilde{M}_t)_{ij})$. Dividing each rotated entry by its true standard deviation and rotating back gives

$$U = Q_L [\widetilde{M}_t \oslash \sqrt{V_{S,t}}] Q_R^\top, \quad C = I.$$

The resulting variance is one in every unit direction, not only along the original coordinates. Thus, SOAP can whiten the update covariance when the input covariance has the assumed Kronecker structure, the rotation Q decorrelates the proposal coordinates, and their variances $V_{S,t}$ are known exactly.

SPlus. Suppose $\text{vec}(M)$ is zero-mean Gaussian and the fixed rotation above gives uncorrelated entries. These entries are then independent, and their signs are independent, equally likely $+1$ and -1 values. Therefore

$$U = Q_L \text{sign}(\widetilde{M}_t) Q_R^\top, \quad C = I.$$

Unlike SOAP, this transformation does not require estimating the coordinate standard deviations.

SPlus vs. SOAP Under Imperfect Decorrelation. The preceding analyses of SOAP and SPlus assume that the rotation decorrelates the entries of M_t , with SOAP additionally having access to the true coordinate standard deviations. We can get further insight by considering an imperfect rotation: fix the estimated Q_L, Q_R before sampling $\text{vec}(M_t) \sim \mathcal{N}(\mathbf{0}, \Sigma)$, and let C be the correlation matrix of the entries of $\widetilde{M}_t = Q_L^\top M_t Q_R$. To characterize how SPlus transforms these correlations, we apply the Gaussian arcsine law (Dirksen et al., 2022), which gives the covariance after taking signs. Rotating back yields

$$C_{\text{SPlus}} = Q \left[\frac{2}{\pi} \arcsin(C) \right] Q^\top, \quad \left| \frac{2}{\pi} \arcsin(\rho) \right| \leq |\rho|. \quad (13)$$

Here $\arcsin$ acts entrywise: the signs have unit variance in the rotated basis, and residual correlations contract, but need not vanish.

With the same rotation and exact coordinate standard deviations, SOAP instead gives $C_{\text{SOAP}} = Q C Q^\top$: coordinate normalization leaves these correlations unchanged. For both methods, the covariance in the rotated basis has unit diagonal. Its squared Frobenius distance to identity is therefore just the sum of the squared off-diagonal entries. Rotating back preserves this distance, yielding

$$\begin{aligned} \|C_{\text{SPlus}} - I\|_F^2 &= \sum_{i \neq j} \left[\frac{2}{\pi} \arcsin(C_{ij}) \right]^2 \\ &\leq \sum_{i \neq j} C_{ij}^2 = \|C_{\text{SOAP}} - I\|_F^2. \end{aligned}$$

Under these assumptions, SPlus produces an update covariance *at least as close to the identity* in Frobenius norm as SOAP, even when the rotation does not perfectly decorrelate the proposal: its sign transformation contracts residual correlations without requiring coordinatewise variance estimates.

Muon. Suppose M is full rank almost surely, and its distribution is unchanged by any orthogonal transformation $M \mapsto OM$ when $m \geq n$ (or $M \mapsto MO$ when $m < n$). For the exact polar transformation with a fixed scale $\gamma > 0$,

$$U = \gamma \text{polar}(M), \quad C = \frac{\gamma^2}{\max(m, n)} I.$$

Thus, Muon produces an isotropic update covariance under strong distributional symmetry assumptions, but its polar transformation does not generally guarantee isotropy.

Relation to MuLoCo MuLoCo studies how Muon’s normalized matrix updates affect pseudo-gradient aggregation (Thérien et al., 2025). It argues that large singular values in individual update matrices contribute to misalignment during aggregation, an issue mitigated by using Muon as the inner optimizer. In contrast, our work examines the covariance structure of Muon’s updates, a distribution-level property that provides a complementary perspective and highlights potential limitations of Muon’s normalization. Furthermore, our error decomposition directly connects this covariance structure to the stochastic component of the error relative to the expected DP update.

Beyond one local step. Equations (10) and (11) apply to the complete displacement after any number of inner optimization steps H . However, if $\Delta_{r,h}$ denotes the displacement at the h -th local step, the covariance of replica r ’s complete displacement is given by

$$C_r = \sum_{h=1}^H \text{Cov}(\Delta_{r,h}) + \sum_{h \neq k} \text{Cov}(\Delta_{r,h}, \Delta_{r,k}).$$

Consequently, whitening individual (one-step) local updates does not by itself whiten the complete pseudo-gradient, as temporal covariances between successive updates can introduce anisotropy. Approximate isotropy of the complete pseudo-gradient can nevertheless be maintained when the aggregate contribution of these temporal covariances is sufficiently small.

4.4 SINGLE-STEP MEAN MISMATCH AND SECOND-MOMENT ESTIMATION

We now consider the single-step setting ($H = 1$), where all replicas share the same initial parameters and optimizer states, and analyze the mean mismatch term of the decomposition introduced in Section 4.2. This eliminates multi-step local drift, allowing us to study the remaining mean mismatch $\mathbf{b}_R$ more easily. At fixed global batch size B , let $\hat{\mathbf{g}}_r := \text{vec}(\mathbf{G}_r)$ denote the independent and identically distributed replica gradients, and define

$$\mathbf{g} := \mathbb{E}[\hat{\mathbf{g}}_r], \quad \bar{\mathbf{g}} := \frac{1}{R} \sum_r \hat{\mathbf{g}}_r, \quad \Sigma_g := \text{Cov}(\bar{\mathbf{g}}). \quad (14)$$

Then $\text{Cov}(\hat{\mathbf{g}}_r) = R\Sigma_g$. For IID examples, $\Sigma_g = \Omega/B$, where Ω is the per-example covariance. Let $T(\hat{\mathbf{g}}; s)$ denote the parameter-update map that takes a gradient $\hat{\mathbf{g}}$ and incoming optimizer state s and returns the corresponding parameter displacement. Then

$$\bar{\Delta} = \frac{1}{R} \sum_r T(\hat{\mathbf{g}}_r; s), \quad \mathbf{d}_1^{\text{DP}} = \mathbb{E}[T(\bar{\mathbf{g}}; s)], \quad \mathbf{b}_R = \alpha_R \mathbb{E}[\bar{\Delta}] - \mathbf{d}_1^{\text{DP}}. \quad (15)$$

In this single-step setting, DP applies the optimizer transformation T to the averaged gradient $\bar{\mathbf{g}}$, whereas DiLoCo applies T independently to each replica’s gradient before averaging the resulting updates. While an affine T commutes with averaging sample-by-sample, sharing the incoming optimizer states does not generally make an adaptive optimizer’s update transformation affine.

We first analyze SPlus, whose sign-based transformation permits explicit expressions for the mean and covariance of its updates.

SPlus: a direct bound on the mean term. Fix the shared SOAP basis before the step; this covers a step between refreshes or a lagged-basis implementation. Momentum is still updated by (1). The average of the replica momenta is exactly the DP momentum from the same incoming state. Let $\tilde{m}_{r,t,j}$ denote entry j of $\text{vec}(\tilde{\mathbf{M}}_{r,t})$, for $j = 1, \dots, d = mn$, and assume

$$\tilde{m}_{r,t,j} = \sigma_j(a_j + \sqrt{R}Z_{r,j}), \quad Z_{r,j} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, 1), \quad \sigma_j > 0. \quad (16)$$

Thus a_j is the mean divided by the standard deviation of the DP momentum coordinate, not its replica-local SNR. We use unit step scale in the following proposition and restore other deterministic scales multiplicatively.

Proposition 2 (SPlus mean and covariance relative to DP). *Under (16), set $\mathbf{a} = (a_j)_{j=1}^d$, $\psi(t) = 2\Phi(t) - 1$, and $c = \sqrt{2/\pi}$. With $\alpha_R = \sqrt{R}$,*

$$\begin{aligned} \mathbf{b}_R &= \mathbf{Q}[\sqrt{R}\psi(\mathbf{a}/\sqrt{R}) - \psi(\mathbf{a})], \\ \|\mathbf{b}_R\|_2^2 &\leq \frac{c^2}{36} \left(1 - \frac{1}{R}\right)^2 \sum_j a_j^6, \end{aligned} \quad (17)$$

$$\text{Cov}(\sqrt{R}\bar{\Delta}) = \mathbf{Q} \text{Diag}(1 - \psi(a_j/\sqrt{R})^2) \mathbf{Q}^\top \preceq \mathbf{I}.$$

Here ψ acts entrywise. In particular, every directional error in (10) is at most $1 + \frac{c^2}{36}(1-R^{-1})^2 \sum_j a_j^6$.

The proof is given in Appendix A.2. For a small vector $\mathbf{a}$, the mean mismatch is third order and its squared contribution is sixth order. At $\mathbf{a} = 0$, the calibrated local aggregate and DP have the same mean and covariance, although their distributions differ. The $\sqrt{R}$ calibration makes the stochastic term order one; it does not retain an additional $1/R$ variance gain at fixed global batch.

Mean mismatch for Adam We now bound the mean mismatch $\mathbf{b}_R$ for Adam. At fixed global batch size, the local gradient variance is R times that of the DP average. Fresh-gradient second moments therefore change the normalization even from a common incoming state. Proposition 3 bounds the resulting **mean-update error**; Appendix A.3 gives the second-moment identities and the Gaussian variance-differentiation argument (Voigtlaender, 2021).

Proposition 3 (Second-moment memory controls a one-step mean bound). *In a fixed parameter coordinate j , let the incoming Adam state be (m, v) , with $v > 0$ and $0 < \beta_2 < 1$. Omit ϵ , bias correction, and the learning rate, and define the complete same-step scalar map*

$$T(x; m, v) = \frac{\beta_1 m + (1 - \beta_1)x}{\sqrt{\beta_2 v + (1 - \beta_2)x^2}}. \quad (18)$$

Suppose $\hat{g}_r \sim \mathcal{N}(g, R\sigma^2)$ independently, so $\bar{g} \sim \mathcal{N}(g, \sigma^2)$. With $\alpha_R = 1$, write $b_{R,j}$ for this coordinate of the bias $\mathbf{b}_R$ from (15). It obeys

$$|b_{R,j}| \leq \frac{(1 - \beta_2)(R - 1)\sigma^2}{2(\beta_2 v)^{3/2}} (|\beta_1 m| + 3(1 - \beta_1)|g|). \quad (19)$$

For SOAP with a basis fixed before the fresh gradients, the same bound applies to the corresponding coordinate of $\mathbf{Q}^\top \mathbf{b}_R$.

This is a finite-noise bound, proved in Appendix A.3.1. For fixed β_2 , the upper bound grows linearly with $R - 1$, motivating an increase in β_2 as the number of replicas grows. Adapting the scaling rule of Marek et al. (2025) to our setting, we can preserve the second-moment EMA’s half-life in terms of local training examples by setting:

$$\beta_2(R) = \beta_2(R_0)^{R_0/R}, \quad 1 - \beta_2(R) = O(R^{-1}). \quad (20)$$

If m, v, g, σ are held fixed and $\beta_2(R)$ follows (20), with $\beta_2(R)$ bounded away from zero, the right-hand side remains bounded rather than growing linearly in R . For each R , both routes in this comparison use the *same* $\beta_2(R)$. **Scaling second-moment memory therefore controls a bound on the Adam mean mismatch as R grows.** Summing the squared coordinate bounds controls $\|\mathbf{b}_R\|_2^2$, and hence the mean term $(\mathbf{q}^\top \mathbf{b}_R)^2$ for every unit $\mathbf{q}$ in (10). The same conclusion holds after SOAP’s orthogonal rotation back to parameter coordinates. This conditional one-step result motivates the matched- β_2 experiments in Section 5.2.

Scale-estimation error at smaller local batches. Beyond the mean mismatch analyzed above, smaller local batches can also increase the variability of second-moment estimates. Indeed, at fixed β_2 , smaller local batches produce noisier gradients, which can increase fluctuations in the estimated second moment $\hat{v}$. For SOAP, these fluctuations in $\hat{v}$ change the coordinatewise normalization and hence the update covariance. SPlus avoids this source of variability by taking signs in its estimated basis rather than dividing by estimated coordinate scales.

5 EMPIRICAL EVALUATION

We first compare replica scaling across inner optimizers (Sec. 5.1), then isolate the effect of second-moment memory on performance (Sec. 5.2), and finally examine the update distributions behind these results (Sec. 5.3). The analysis connects the observed scaling to the two terms in Equation (10): mean mismatch and stochastic aggregation error.

Experimental setup. Our main experiments train 178M- and 512M-parameter LLaMA models on DCLM (Li et al., 2024) with sequence length 2048. We evaluate performance on a held-out 100M-token validation split. The 178M runs use a fixed global batch of 524K tokens and approximately 3.55B training tokens; the 512M runs use 2M tokens per global batch and 10.25B-token budget. Unless stated otherwise, replicas take $H = 30$ local steps between aggregations and all local optimization methods use a Nesterov SGD outer optimizer. Appendix B records the architecture, training budgets, and result-selection protocol.

5.1 COMPARING INNER OPTIMIZERS IN REPLICA SCALING

We compare SOAP, SPlus, Muon, AdamW, and Shampoo as inner optimizers for DiLoCo while scaling the number of replicas. Figures 2 and 3 show both absolute validation loss and the percentage change from each optimizer’s own DP reference (Thérien et al., 2025). The latter allows making a direct comparison between inner optimizers as it relates to their impact on worker scaling without being confounded by absolute performance differences of the optimizers (e.g., accounting for the fact that Muon has lower absolute loss than AdamW).

178M Experiment Scale At 178M scale (Fig 2), we observe, as in Thérien et al. (2025), that Muon has better replica scaling than AdamW, not just DP performance. SOAP has both stronger DP performance and significantly improved replica scaling compared to Muon and Adam. SPlus has the strongest worker scaling, leading to the smallest increase over its own DP reference and the lowest observed loss at large R . At $R = 64$, the loss change is +3.1% for SPlus and +4.8% for SOAP, compared with +8.5% for Muon and +11% for AdamW. Shampoo improves on AdamW but remains behind SOAP and SPlus. As discussed in our work, it does not assure an approximately whitened update unlike SOAP and SPlus. Thus, both the normalization and the coordinates in which it is applied matter empirically. Section 5.3 examines how this ordering relates to the covariance of the returned updates. Interestingly, at 178M, $R = 8$ SPlus actually improves performance over its DP baseline. This is likely because at $R > 1$ it can mimic a majority vote-based technique (Bernstein et al., 2018).

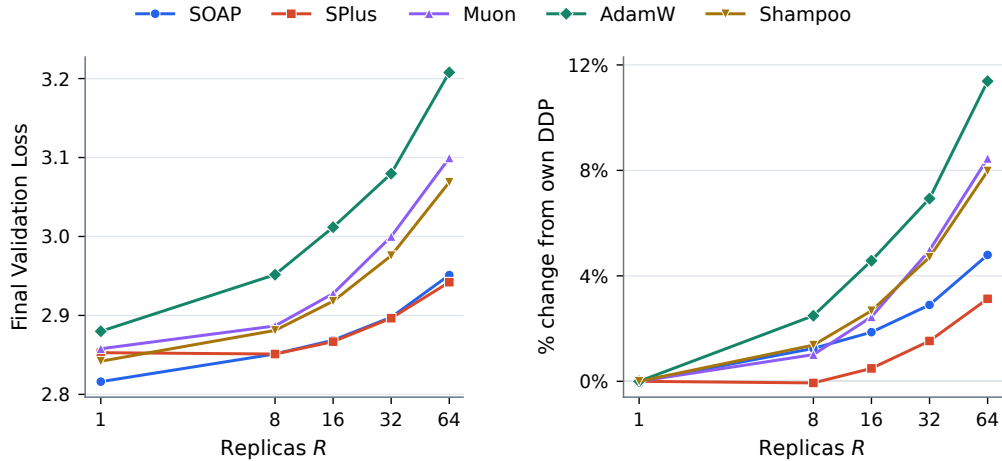


Figure 2: **178M replica scaling.** $R = 1$ reports DP performance, while $R > 1$ reports the performance of local optimization with synchronization interval $H = 30$. Left: final validation loss. Right: change in loss from the same optimizer’s own DP ($R=1$) baseline. We observe that matrix-based optimizers have stronger worker scaling than AdamW and that SPlus achieves the strongest scaling of all relative to its DP baseline.

512M Experiment Scale In Figures 3, we observe a similar pattern. SOAP is, once again, the strongest DP reference and has the lower loss through $R = 32$. SPlus overtakes it at $R = 64$ and $R = 96$. At $R = 96$, SPlus’ los change is +2.1% **above its own DP loss**, versus +4.4% for SOAP, +7.7% for Muon, and +12% for AdamW. The crossover is consistent across model scales: SOAP’s DP performance is stronger, whereas SPlus degrades less as replicas are scaled and the local batch size shrinks. Both SPlus and SOAP show drastically better scaling than Adam and Muon, making them practical alternatives when training with high worker counts.

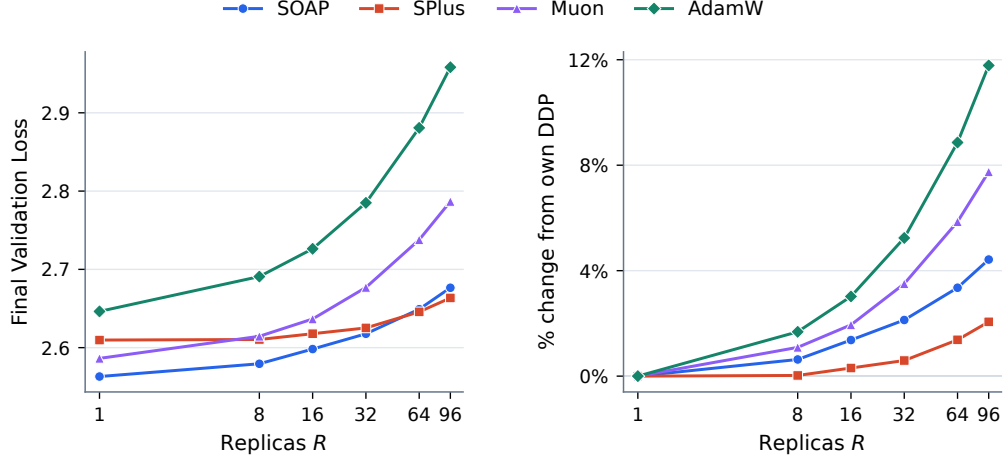


Figure 3: **512M replica scaling.** $R = 1$ reports DP performance, while $R > 1$ reports the performance of local optimization with synchronization interval $H = 30$. Left: final validation loss. Right: change in loss from the same optimizer’s own DP ($R=1$) baseline. As in Fig. 2, we observe that matrix-based optimizers have stronger worker scaling than AdamW and that SPlus achieves the strongest scaling of all relative to its DP baseline.

Table 1: **178M final validation loss.** DP uses $R = 1$; other runs use $H = 30$. Best in each column is bold.

Optimizer	Replicas R				
	DP	8	16	32	64
SOAP	2.816	2.851	2.869	2.898	2.951
SPlus	2.853	2.851	2.867	2.897	2.942
Muon	2.858	2.887	2.928	3.000	3.099
AdamW	2.880	2.952	3.012	3.080	3.208
Shampoo	2.842	2.881	2.918	2.976	3.069

Table 2: **512M final validation loss.** DP uses $R = 1$; other runs use $H = 30$. Best in each column is bold.

Optimizer	Replicas R					
	DP	8	16	32	64	96
SOAP	2.563	2.579	2.598	2.618	2.649	2.676
SPlus	2.610	2.610	2.618	2.625	2.646	2.663
Muon	2.586	2.615	2.637	2.677	2.738	2.787
AdamW	2.646	2.691	2.726	2.785	2.881	2.958

Comparing results across 178M and 512M Prior work has demonstrated that the worker scaling performance of AdamW and Muon improves as data and model scale are increased (Thérien et al., 2025; Charles et al., 2025). In Figure 1, we compare the performance of SPlus, SOAP, AdamW, and Muon when scaling replicas. We also observe that scale improves replica scaling across all optimizers can significantly benefit from this including SOAP, SPlus, and Muon.

Isolating the Effect of Local Steps: Replica-scaling effects are visible at $H = 1$, before multi-step parameter drift can occur! Unlike SGD, in adaptive optimizers, the divergence to standard DP training occurs even with a single step of local optimization (due to the non-linear map of gradients to update). Indeed, the arguments regarding covariance structure in Sec. 4 apply directly to this case. To isolate the effects of local drift when taking multiple steps from the impact of the adaptive optimizer update, we perform scaling experiments using a single local update. We also remove the impacts of the outer optimizer by using a standard SGD outer optimizer with a learning rate of 1 (equivalent

to parameter averaging). Table 3 reports the results of this experiment. We observe similar results to the multi-step setting: Adam, Muon, and Shampoo have poorer replica scaling, while SOAP and SPlus remain substantially more robust. This is the setting in which Section 4.4 isolates the effect of applying the optimizer before gradient averaging. We also observe that performance (even in terms of scaling with respect to DP) is far more impacted by the choice of optimizer and the number of replicas than the number of local steps. This suggests the impact of the local optimizer on the update and subsequent aggregation can be more significant than the parameter drift often studied in local optimization (Stich, 2019; Karimireddy et al., 2020).

Table 3: **178M performance using $H=1$ (single-step) local updates.** We report the final validation loss for each method. To isolate the effect of the adaptive optimizer’s update, we use pure parameter averaging at each step and persistent local optimizer states. All methods train for 6,775 steps. The DP, $R = 8$, and $R = 64$ methods are directly tuned, while the $R = 16, 32$ configurations use interpolated hyperparameters rather than separate full sweeps. The best-performing method in each column is bolded.

Inner optimizer	DP	$R = 8$	$R = 16$	$R = 32$	$R = 64$
SOAP	2.816	2.842	2.864	2.901	2.958
SPlus	2.853	2.837	2.853	2.873	2.912
Muon	2.858	2.979	3.073	3.230	3.417
AdamW	2.880	2.932	2.969	3.050	3.157
Shampoo	2.842	2.899	2.947	3.002	3.097

Interestingly, Muon is less robust here than at $H = 30$ and is worse than AdamW at the larger replica counts. Its strong $H = 30$ results therefore do not transfer unchanged to pure one-step averaging. We hypothesise that this may be due to differences in their outer optimizer, which could be more favourable for some methods and less for others.

5.2 SECOND-MOMENT MEMORY AND REPLICA SCALING

Following Sec. 4.4, we next study a practical consequence of smaller local batches for Adam-based methods. Equation (A.6) shows how local gradient variance enters the second-moment discrepancy. The token-half-life rule in Equation (20) increases β_2 as R grows, preserving the number of tokens represented by the running average. In this section, we study the impact of applying this and, in particular, evaluate how it impacts conclusions in prior work (Thérien et al., 2025; Horoi et al., 2026). This rule is already incorporated in the main results in Sec. 5.1. Here we evaluate both the Adam-based DiLoCo optimizer and IsoLoCo which is a modification of the outer optimizer. We incorporate it into our analysis as it is based on the Adam inner optimizer, which can benefit from appropriate β_2 tuning. We note that in the prior section’s experiments, to ensure the strongest baseline, this rule is already used by construction in all optimizers with a second moment. This section, on the other hand, emphasizes the improvement compared to existing literature, which does not scale β_2 when scaling R Charles et al. (2025); Horoi et al. (2026); Sarfi et al. (2025)

To empirically measure the importance of second moment estimation highlighted in Sec. 4.4, we compared the performance of optimizers using a fixed β_2 and tuning it at every replica scale. Figure 4 reports the final validation loss of DiLoCo, IsoLoCo, and MuLoCo under these strategies. We note that while IsoLoCo and DiLoCo both apply AdamW to each tensor, MuLoCo only applies it to non-hidden tensors. We observe all optimizers improve from tuning β_2 as replicas increase, but IsoLoCo and DiLoCo benefit more. Under β_2 tuning, we find that IsoLoCo, despite using an AdamW inner optimizer, can even outperform MuLoCo at large worker counts.

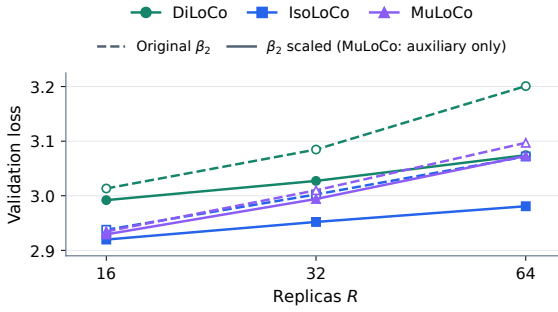


Figure 4: **The effect of tuning AdamW’s β_2 in DiLoCo, IsoLoCo, and MuLoCo.** We report the performance of each method at 178M scale when fixing (dashed lines) and tuning (solid lines) β_2 at each worker count. While IsoLoCo and DiLoCo both apply AdamW to each tensor, MuLoCo only applies it to non-hidden layers. We observe all optimizers improve from this change, but IsoLoCo and DiLoCo benefit significantly more, revealing that IsoLoCo can outperform MuLoCo at larger worker counts when β_2 is well tuned.

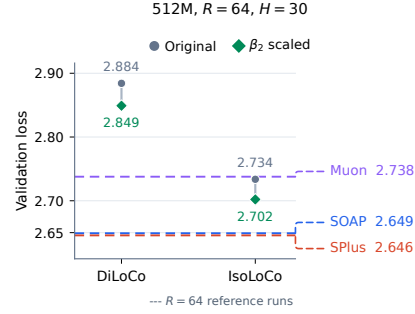


Figure 5: **Tuning β_2 for larger model and replica.** We ablate using fixed β_2 vs scaling at high replica count ($R=64$) and 512M model scale. We observe that performance consistently improves for DiLoCo and IsoLoCo. Notably IsoLoCo outperforms Muon inner optimizer (MuLoCo).

We also confirm that this scaling rule works well at 512M in Figure 5. Both methods use the published $R = 8$ hyperparameters without retuning. At 512M and $R = 64$, changing only β_2 reduces DiLoCo loss from 2.884 to 2.849 and IsoLoCo loss from 2.734 to 2.702, approximately 1.2% in each case. Second-moment memory is thus an effective control on replica scaling, including for IsoLoCo, whose outer optimizer does not remove this inner AdamW sensitivity. Notably, after the relevant control, IsoLoCo, which uses an AdamW inner optimizer, can outperform MuLoCo, while only doing the spectral decomposition during the outer step.

Discussion of Practical Implications Our results suggest that local optimization methods can be deployed across a large number of replicas (50+) while maintaining minimal performance degradation of $< 3\%$ when utilizing SOAP or SPlus. Notably, these methods consume more memory than Muon due to the need to maintain L and R , however, scalable recipes for SOAP have been studied (Khona et al., 2026), demonstrating that it can perform well at very large scale. A particularly relevant setting for our work is cross-data center training where each replica has a considerable number of chips that can be used to shard the additional optimizer states and perform the optimizer step in parallel. At 178M and 512M scales, our results suggest that at a smaller number of replicas SOAP is the preferred inner optimizer, while at a large number of replicas, SPlus performs best. Another observation of our work is that the recently proposed IsoLoCo outer optimizer, which has nearly the same overhead as AdamW inner optimizer-based DiLoCo, can perform very effectively at a large number of replicas simply by adjusting its β_2 according to the local batch size. When doing so, we show that it outperforms MuLoCo. This can make it an excellent candidate for settings where each replica has limited memory or existing infrastructure and training recipe are already well designed for Adam.

5.3 ANALYSIS: COVARIANCE STRUCTURE AND MEAN MISMATCH

We empirically analyze the covariance structure and mean mismatch discussed in Section 4 for the different optimizer classes.

Empirically Measuring Whitening Using the 178M training setting and beginning with checkpoints at outer round 75, we empirically estimate how close to whitened the covariance of the pseudo-gradient is for different inner optimizers. We perform 64 instantiations of inner training loop starting with the same checkpoint and inner optimizer state, thus only varying the data used.

Subsequently we use pairs to construct an estimator for the following quantities:

$$\tilde{\mathbf{C}}_h := \frac{1}{R^2} \sum_{r=1}^R \mathbf{C}_{r,h}, \quad a_h = \left(\frac{d \operatorname{tr}(\tilde{\mathbf{C}}_h^2)}{\operatorname{tr}(\tilde{\mathbf{C}}_h)^2} - 1 \right)^{1/2}. \quad (21)$$

Here d is the probed dimension; $a_h = 0$ means isotropy, and larger values indicate greater anisotropy, independently of overall noise magnitude. The constructed $\tilde{\mathbf{C}}_h$ equals the covariance of the averaged update when cross-worker noise covariances vanish, conditional on the starting parameters and optimizer states. Appendix C.1 gives the estimators and exact replay protocol. We chose this particular quantity as a proxy for isotropy as the trace and square trace have efficient estimators available in the statistical literature. The results are shown in Table 4 below.

Table 4: Update-noise anisotropy at $R = 64$, round 75, using all 64 replicas and local batch four. Lower is more isotropic.

Optimizer	$a_1 \downarrow$	$a_{30} \downarrow$
AdamW	35	11
Muon	3.8	3.3
SOAP	2.3	2.5
SPlus	0.43	2.4

Table 5: Mean mismatch is smaller than stochastic error in the detected high-noise directions, and particularly small for SPlus and SOAP compared with AdamW.

Optimizer	Mean mismatch ($\times 10^{-6}$)	Stochastic error ($\times 10^{-6}$)	Mean / noise (%)
SPlus	.044	.91	7.9
SOAP	.12	1.4	7.5
Muon	.11	1.1	14
AdamW	5.4	19	19

We observe that *covariance structure broadly follows the scaling results*. with optimizers that have better replica scaling having better covariance structure. The anisotropy differences are already pronounced after one local step, before parameter drift across replicas can affect subsequent gradients. a_1 ranges from 35 for AdamW to 0.43 for SPlus, with Muon and SOAP between them. After 30 steps, SOAP and SPlus remain less anisotropic than Muon and AdamW, broadly following their replica-scaling performance. Interestingly, Muon is significantly more isotropic than Adam, helping to explain its improved scaling Thérien et al. (2025). The next measurements examine the curvature of high-variance noise directions, connecting this shape comparison to optimization.

Noise dominates mean mismatch along high-noise directions. Finally, we compare the two terms of Equation (10) along directions selected for high update variance. For each optimizer, all replicas and its DP reference start from the same checkpoint weights and replica 0’s saved optimizer state. They run $H = 30$ steps on paired data: DP uses the union of the local batches at each step. We analyze the 63 hidden attention and MLP weight matrices separately, excluding embeddings, the output head, and other auxiliary parameters.

For each matrix, 16 repetitions fit a scalar α_R and a leading noise direction $\hat{\mathbf{q}}$; 24 additional repetitions estimate the directional mean mismatch $(\hat{\mathbf{q}}^\top \mathbf{b}_R)^2$ and stochastic error $\alpha_R^2 \hat{\mathbf{q}}^\top \mathbf{C}_{\text{agg}} \hat{\mathbf{q}}$. Both terms are divided by the squared norm of that matrix’s DP mean update. This per-matrix calibration applies the decomposition to each matrix separately; it is not a change to the training learning rates. Appendix C.2 defines the estimators, including the paired finite-sample correction.

We observe *the mean-mismatch term is only 7.5–19% of the stochastic term on average along these directions*. This supports controlling the stochastic term in Equation (10) which is the focus of our analysis in Section 4, and allow to better confirm that the covariance structure of SPlus and SOAP lead to their improved replica scaling.

Extreme noise also lies in high-curvature directions. We have seen bounding the maximum noise direction of the covariance as in Equation 11 can reduce the maximum discrepancy to the DDP direction, in particular since the above empirical experiments suggest it dominates the bound on the error with respect to the DDP target. We next ask whether directions carrying extreme noise variance are directly responsible for poor performance of methods such as AdamW which do not bound them well. We do this by estimating these direction and then the underlying curvature. At each method’s own checkpoint for outer step 75, we fit a unit direction $\hat{\mathbf{q}}$ that maximizes sample variance over 64 paired one-step update differences. With $\mathcal{H}$ the loss Hessian on 64 held-out sequences, Table 6

reports

$$\kappa(\hat{\mathbf{q}}) = \frac{\hat{\mathbf{q}}^\top \mathbf{H} \hat{\mathbf{q}}}{\frac{1}{16} \sum_{j=1}^{16} \mathbf{v}_j^\top \mathbf{H} \mathbf{v}_j},$$

where the $\mathbf{v}_j$ are independent random unit vectors in the same six-matrix subspace. A ratio of 100 means 100 times the average random-direction *curvature*, not noise variance.

Appendix C.3 details the estimator used. We observe that Adam and Muon have very high curvature in the directions with the most noise compared to average directions, in particular as we scale the replica size, while SOAP and SPlus are able to effectively redistribute noise in any particular direction due to their whitening. High noise along high curvature directions can be particularly problematic as small errors can be very damaging for the loss in high curvature regions. This gives an additional mechanism for understanding why whitening (which redistributes error more uniformly along all directions) can be particularly important in this setting.

Table 6: Loss curvature along a fitted high-noise direction, relative to random directions. Ranges span replicas 0 and 1, not confidence intervals; local batch $b = 256/R$.

Optimizer	Curvature ratio $\kappa(\hat{\mathbf{q}})$	
	$R = 8$	$R = 64$
AdamW	86–170	460–730
Muon	6.4–7.7	47–100
SOAP	3.1–3.2	4.2–5.2
SPlus	1.5–1.6	1.5

6 CONCLUSION

We have shown that differences to the DP pseudogradient can be seen to result from two primary sources of error: mean mismatch and stochastic error. Building on these insights, we showed that covariance structure in optimizers like AdamW is very important. In our empirical study, we establish the performance of different inner optimizers (SOAP, SPlus, Muon, AdamW), which have more and less appealing properties for reducing stochastic error according to our theory. We find that SPlus reaches the strongest performance at higher worker counts, and the best overall scaling with respect to DP, while SOAP reaches the best DP performance and seems on trend to become the best inner optimizer at scale. We then directly test our theoretical claims by calculating the mean mismatch and stochastic error for each optimizer, showing that these quantities are directly correlated with the observed performance trends.

ACKNOWLEDGEMENTS

We acknowledge support from NSERC Discovery RGPIN-2021-04104 and compute resources from Digital Research Alliance Canada.

REFERENCES

- A. C. Aitken. On least squares and linear combinations of observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1935. doi:10.1017/S0370164600014346.
- Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 560–569, 2018. URL <https://proceedings.mlr.press/v80/bernstein18a.html>.
- Zachary Charles, Gabriel Teston, Lucio Dery, Keith Rush, Nova Fallen, Zachary Garrett, Arthur Szlam, and Arthur Douillard. Communication-efficient language model training scales reliably and robustly: Scaling laws for DiLoCo. In *Advances in Neural Information Processing Systems*, 2025. URL <https://arxiv.org/abs/2503.09799>.
- Sjoerd Dirksen, Johannes Maly, and Holger Rauhut. Covariance estimation under one-bit quantization. *The Annals of Statistics*, 50(6):3538–3562, 2022. doi:10.1214/22-AOS2239. URL <https://arxiv.org/abs/2104.01280>.

- Arthur Douillard, Qixuan Feng, Andrei A. Rusu, Rachita Chhaparia, Yani Donchev, Adhiguna Kuncoro, Marc’Aurelio Ranzato, Arthur Szlam, and Jiajun Shen. DiLoCo: Distributed low-communication training of language models. arXiv:2311.08105, 2023. URL <https://arxiv.org/abs/2311.08105>.
- Arthur Douillard, Yanislav Donchev, Keith Rush, Satyen Kale, Zachary Charles, Zachary Garrett, Gabriel Teston, Dave Lacey, Ross McIlroy, Jiajun Shen, Alexandre Ramé, Arthur Szlam, Marc’Aurelio Ranzato, and Paul Barham. Streaming DiLoCo with overlapping communication: Towards a distributed free lunch. arXiv:2501.18512, 2025. URL <https://arxiv.org/abs/2501.18512>.
- Kevin Frans, Sergey Levine, and Pieter Abbeel. A stable whitening optimizer for efficient neural network training, 2025. URL <https://arxiv.org/abs/2506.07254>.
- Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1842–1850, 2018. URL <https://proceedings.mlr.press/v80/gupta18a.html>.
- Stefan Horoi, Benjamin Thérien, Guy Wolf, and Eugene Belilovsky. Can model merging improve aggregation in DiLoCo? arXiv:2607.03011, 2026. URL <https://arxiv.org/abs/2607.03011>.
- Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. Technical article, 2024. URL <https://kellerjordan.github.io/posts/muon/>.
- Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5132–5143, 2020. URL <https://proceedings.mlr.press/v119/karimireddy20a.html>.
- Agan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018. doi:10.1080/00031305.2016.1277159. URL <https://arxiv.org/abs/1512.00809>.
- Mikhail Khona, Aditya Vavre, Boxiang Wang, Deyu Fu, Hao Wu, Mike Chrzanowski, Bryan Catanzaro, Dheevatsa Mudigere, Jeff Pool, Michael Lightstone, Mohammad Shoeybi, Mostofa Patwary, Nima Tajbakhsh, and Tijmen Blankevoort. SOAP, muon, and beyond: Pushing LLM pretraining scales. arXiv:2607.20548, 2026. URL <https://arxiv.org/abs/2607.20548>.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL <https://arxiv.org/abs/1412.6980>.
- Jeffrey Li et al. DataComp-LM: In search of the next generation of training sets for language models. arXiv:2406.11794, 2024. URL <https://arxiv.org/abs/2406.11794>.
- Wu Lin, Scott C. Lowe, Felix Dangel, Runa Eschenhagen, Zikun Xu, and Roger B. Grosse. Understanding and improving shampoo and SOAP via Kullback–Leibler minimization. In *International Conference on Learning Representations*, 2026. URL <https://arxiv.org/abs/2509.03378>.
- Bo Liu et al. Asynchronous local-SGD training for language modeling. arXiv:2401.09135, 2024. URL <https://arxiv.org/abs/2401.09135>.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL <https://arxiv.org/abs/1711.05101>.
- Sadhika Malladi, Kaifeng Lyu, Abhishek Panigrahi, and Sanjeev Arora. On the SDEs and scaling rules for adaptive gradient algorithms. In *Advances in Neural Information Processing Systems*, 2022. URL <https://arxiv.org/abs/2205.10287>.

- Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL <https://arxiv.org/abs/2502.04959>.
- Martin Marek, Sanae Lotfi, Aditya Somasundaram, Andrew Gordon Wilson, and Micah Goldblum. Small batch size training for language models: When vanilla SGD works, and why gradient accumulation is wasteful, 2025. URL <https://arxiv.org/abs/2507.07101>.
- H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, 2017. URL <https://proceedings.mlr.press/v54/mcmahan17a.html>.
- Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. URL <https://arxiv.org/abs/2003.00295>.
- Matthew Reed and Mohamed Baioumy. SPARTA: Distributed training with sparse parameter averaging. EXO Labs, 2025. URL <https://openreview.net/forum?id=stFPf3gzq1>. See also the OpenReview preprint “Improving the Efficiency of Distributed Training using Sparse Parameter Averaging”.
- Amir Sarfi et al. Communication efficient LLM pre-training with SparseLoCo. arXiv:2508.15706, 2025. URL <https://arxiv.org/abs/2508.15706>.
- Sebastian U. Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2019. URL <https://arxiv.org/abs/1805.09767>.
- Benjamin Thérien, Xiaolong Huang, Aaron Defazio, Irina Rish, and Eugene Belilovsky. MuLoCo: Muon is a practical inner optimizer for DiLoCo, 2025. URL <https://arxiv.org/abs/2505.23725>.
- Felix Voigtlaender. A general version of Price’s theorem: A tool for bounding the expectation of nonlinear functions of Gaussian random vectors. *Journal of Theoretical Probability*, 34:1474–1485, 2021. doi:10.1007/s10959-020-01017-w.
- Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. SOAP: Improving and stabilizing shampoo using adam. In *International Conference on Learning Representations*, 2025. URL <https://arxiv.org/abs/2409.11321>.
- Jianyu Wang, Rudrajit Das, Gauri Joshi, Satyen Kale, Zheng Xu, and Tong Zhang. On the unreasonable effectiveness of federated averaging with heterogeneous data. *Transactions on Machine Learning Research*, 2024. URL <https://arxiv.org/abs/2206.04723>.
- Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. arXiv:2509.02046, 2025. URL <https://arxiv.org/abs/2509.02046>.
- Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7184–7193, 2019. URL <https://proceedings.mlr.press/v97/yu19d.html>.

Appendix A collects the supporting theoretical results and proofs. Appendix B records the training setup, and Appendix C gives the analysis protocols and estimators.

A SUPPORTING THEORETICAL RESULTS

A.1 MUON COVARIANCE UNDER ROTATIONAL PROPOSALS

Let $p = \min(m, n)$ and $d_{\max} = \max(m, n)$. If a full-rank proposal has SVD $M = \tilde{U} \text{diag}(\sigma_i) \tilde{V}^\top$, the ideal Muon update with prescribed matrix-size scale $\gamma > 0$ is

$$U_{\text{Muon}} = \gamma \text{polar}(M) = \gamma \tilde{U} \tilde{V}^\top. \quad (\text{A.1})$$

The polar map discards the realized singular values and keeps the polar factor. In particular, $\text{polar}(aM) = \text{polar}(M)$ for every $a > 0$, so a positive multiplicative change to the entire proposal leaves the update unchanged.

Proposition 4 (Muon covariance under rotational proposals). *Suppose M_r is full rank almost surely. If $m \geq n$, suppose $OM_r \stackrel{d}{=} M_r$ for every $O \in O(m)$; if $m < n$, suppose $M_r O \stackrel{d}{=} M_r$ for every $O \in O(n)$. Either one-sided condition suffices when $m = n$. Then*

$$\mathbb{E}[U_{\text{Muon},r}] = \mathbf{0}, \quad \text{Cov}(\text{vec}(U_{\text{Muon},r})) = \frac{\gamma^2}{d_{\max}} \mathbf{I}_{mn}. \quad (\text{A.2})$$

Consequently, independent replicas have a uniformly averaged update covariance $\gamma^2 \mathbf{I}_{mn} / (d_{\max} R)$, even when their proposal singular-value distributions differ.

Proof. Consider $m \geq n$ and write the columns of $Q_r = \text{polar}(M_r)$ as $q_{r,1}, \dots, q_{r,n}$. Equivariance of the polar map and left invariance of the proposal law imply $OQ_r \stackrel{d}{=} Q_r$ for every $O \in O(m)$. Taking $O = -I$ gives $\mathbb{E}[Q_r] = \mathbf{0}$. Moreover, $\mathbb{E}[q_{r,j} q_{r,k}^\top]$ commutes with every orthogonal matrix and must therefore equal $a_{jk} \mathbf{I}_m$. Since $Q_r^\top Q_r = \mathbf{I}_n$,

$$ma_{jk} = \mathbb{E}[q_{r,k}^\top q_{r,j}] = \mathbf{1}_{\{j=k\}}. \quad (\text{A.3})$$

The column blocks of $\mathbb{E}[\text{vec}(Q_r) \text{vec}(Q_r)^\top]$ are consequently $\mathbf{1}_{\{j=k\}} \mathbf{I}_m / m$, which proves Equation (A.2) because $d_{\max} = m$. The proof for $m < n$ applies the same argument to the rows, using right invariance and $Q_r Q_r^\top = \mathbf{I}_m$. $\square$

A.2 PROOF OF PROPOSITION 2 (SPLUS TO DDP COMPARISON)

Proof of Proposition 2. Write $\psi(a) = 2\Phi(a) - 1$ and $c = \sqrt{2/\pi}$. For one coordinate, set $X_r = \sigma(a + \sqrt{R}Z_r)$ and $\bar{X} = R^{-1} \sum_r X_r$. Then

$$\Pr(X_r > 0) = \Phi(a/\sqrt{R}), \quad \mathbb{E}[\text{sign}(X_r)] = \psi(a/\sqrt{R}), \quad \mathbb{E}[\text{sign}(\bar{X})] = \psi(a).$$

In the shared-state model the last expression is exactly the mean of the signed DDP momentum coordinate, because momentum is affine before the sign. The expectation of the local average equals the single-replica expectation. For $a \geq 0$,

$$\sqrt{R}\psi(a/\sqrt{R}) - \psi(a) = c \int_0^a \left(e^{-u^2/(2R)} - e^{-u^2/2} \right) du. \quad (\text{A.4})$$

The integrand is nonnegative and at most $u^2(1 - R^{-1})/2$, by the unit Lipschitz bound for e^{-x} on $x \geq 0$. Integration and odd symmetry give

$$\left| \sqrt{R}\psi(a/\sqrt{R}) - \psi(a) \right| \leq \frac{c}{6} (1 - R^{-1}) |a|^3.$$

Squaring and summing over coordinates is valid because Q is orthogonal. Each sign has variance $1 - \psi(a_j/\sqrt{R})^2$. Independence of projected coordinates and of replicas gives the covariance in (17). Finally use Proposition 1. $\square$

The DDP covariance is $Q \text{Diag}(1 - \psi(a_j)^2) Q^\top$. For $R \geq 1$ the calibrated local covariance minus this matrix is positive semidefinite, with operator norm at most $c^2 \max_j a_j^2$. Thus mean agreement to first order should not be read as a variance improvement over signed DDP. At zero signal the covariances agree exactly.

A.3 ADAM/SOAP SECOND MOMENTS AND THE MEAN COMPARISON

Condition on common incoming optimizer states and parameters, as in Section 4.4. Let t denote the fresh-gradient step. Averaging squared gradients is not the same as squaring their average: the difference is the scatter of the replica gradients. Define $\mathbf{S}_R = R^{-1} \sum_r (\hat{\mathbf{g}}_r - \bar{\mathbf{g}})(\hat{\mathbf{g}}_r - \bar{\mathbf{g}})^\top$. For the second moments in (2) and (5),

$$\begin{aligned} \frac{1}{R} \sum_r \mathbf{v}_{A,r,t} - \mathbf{v}_{A,\text{DDP},t} &= (1 - \beta_2) \text{diag}(\mathbf{S}_R), \\ \frac{1}{R} \sum_r \mathbf{v}_{S,r,t} - \mathbf{v}_{S,\text{DDP},t} &= (1 - \beta_2) \text{diag}(\mathbf{Q}^\top \mathbf{S}_R \mathbf{Q}). \end{aligned} \quad (\text{A.5})$$

The second identity assumes a common basis fixed before the fresh gradients; $\mathbf{v}_A, \mathbf{v}_S$ are the vectorized buffers. Thus locally updated second moments already differ from DDP before any parameter drift.

Under the main-text batch model, $\text{Cov}(\hat{\mathbf{g}}_r) = R \Sigma_g$ and the replica gradients are independent, so Σ_g is the covariance of the DDP-averaged gradient. At fixed B , taking expectations in (A.5) gives

$$\mathbb{E} \left[\frac{1}{R} \sum_r \mathbf{v}_{A,r,t} - \mathbf{v}_{A,\text{DDP},t} \right] = (1 - \beta_2)(R - 1) \text{diag}(\Sigma_g). \quad (\text{A.6})$$

SOAP has the same expected identity in its fixed basis.

Expanding squares around the sample mean gives

$$\frac{1}{R} \sum_r \hat{\mathbf{g}}_r^{\odot 2} - \bar{\mathbf{g}}^{\odot 2} = \frac{1}{R} \sum_r (\hat{\mathbf{g}}_r - \bar{\mathbf{g}})^{\odot 2} = \text{diag}(\mathbf{S}_R).$$

Multiplying by $1 - \beta_2$ proves the Adam identity in (A.5); applying it to $\mathbf{Q}^\top \hat{\mathbf{g}}_r$ proves the SOAP identity when $\mathbf{Q}$ is fixed and common. Under (14), $\mathbb{E}[\mathbf{S}_R] = (R - 1) \Sigma_g$, which proves (A.6).

For a scalar smooth map f and $X_t = g + \sqrt{t} \sigma Z$, Gaussian variance differentiation (Voigtlaender, 2021) gives the exact identity

$$\frac{d}{dt} \mathbb{E}[f(X_t)] = \frac{\sigma^2}{2} \mathbb{E}[f''(X_t)], \quad \mathbb{E}[f(X_R)] - \mathbb{E}[f(X_1)] = \frac{\sigma^2}{2} \int_1^R \mathbb{E}[f''(X_t)] dt. \quad (\text{A.7})$$

Differentiation under the integral is justified, for example, by bounded first and second derivatives and Gaussian integrability; the Adam map below satisfies the required conditions when the incoming second moment is positive.

A.3.1 FINITE-NOISE MEAN BOUND AND SECOND-MOMENT MEMORY

Proof of Proposition 3. For this proof only, abbreviate $p = \beta_1 m$, $k = 1 - \beta_1$, $a_0 = \beta_2 v > 0$, and $b_0 = 1 - \beta_2 > 0$. Write

$$f(x) = p h(x) + k \ell(x), \quad h(x) = (a_0 + b_0 x^2)^{-1/2}, \quad \ell(x) = x(a_0 + b_0 x^2)^{-1/2}.$$

Direct differentiation gives

$$h''(x) = \frac{b_0(2b_0 x^2 - a_0)}{(a_0 + b_0 x^2)^{5/2}}, \quad \ell'''(x) = \frac{-3a_0 b_0(a_0 - 4b_0 x^2)}{(a_0 + b_0 x^2)^{7/2}}. \quad (\text{A.8})$$

For $y \geq 0$, the suprema of $|2y - 1|/(1 + y)^{5/2}$ and $|1 - 4y|/(1 + y)^{7/2}$ are both one. The first follows by differentiation on either side of $y = 1/2$, and the second on either side of $y = 1/4$; their interior positive extrema have value less than one. Substituting $y = b_0 x^2/a_0$ gives

$$\|h''\|_\infty \leq b_0/a_0^{3/2}, \quad \|\ell'''\|_\infty \leq 3b_0/a_0^{3/2}.$$

Since ℓ'' is odd and Z is symmetric, $\mathbb{E}[\ell''(\sqrt{t}\sigma Z)] = 0$. The mean-value bound therefore implies

$$|\mathbb{E}[\ell''(g + \sqrt{t}\sigma Z)]| \leq 3b_0|g|/a_0^{3/2}.$$

Thus, uniformly in $t > 0$,

$$|\mathbb{E}[f''(g + \sqrt{t}\sigma Z)]| \leq \frac{b_0}{a_0^{3/2}} (|p| + 3k|g|).$$

Integrating (A.7) over $1 \leq t \leq R$ gives (19). This proof does not replace the random denominator by its expectation and does not require a small-noise limit. $\square$

Scaling rule A local example or token is downweighted by β_2 per local step. Keeping its decay per fixed number of local examples unchanged gives $\log \beta_2(R)/(B/R) = \log \beta_2(R_0)/(B/R_0)$, proving (20). Write $c_0 = -R_0 \log \beta_2(R_0) > 0$. Then $1 - \beta_2(R) = 1 - e^{-c_0/R} \leq c_0/R$, so $(R-1)(1 - \beta_2(R)) \leq c_0$. This controls the upper bound with fixed incoming state; it neither forces the bias to zero nor proves that its actual magnitude decreases monotonically with β_2 .

A.4 BASIS AND SCALE ERRORS

SOAP whitening requires both a basis that reduces correlations and coordinate scales that normalize the remaining variances. We first separate these two sources of error in the covariance of the returned update, then show how they affect covariance matching and the stochastic term in Equation (10).

Conditioning and scope. The covariance calculations below concern centered proposals with bases and denominators fixed before the proposal is drawn. They describe an idealized one-step optimizer core, omitting a common learning-rate factor and the numerical stabilizer. The ideal denominators estimate the centered variance of this proposal. In practical SOAP, the denominator instead uses gradient second moments, whereas the numerator contains momentum; refreshing either state from the same gradient also couples the map to its input. The identities below therefore describe the frozen-state model, not the full fresh-gradient Adam or SOAP step.

A.4.1 SCALE ERRORS AND CORRELATIONS LEFT IN THE ESTIMATED BASIS

Let worker r have a centered matrix proposal $M_r \in \mathbb{R}^{m \times n}$, and set $\mathbf{x}_r = \text{vec}(M_r)$ and $\Sigma_r = \text{Cov}(\mathbf{x}_r) \succ \mathbf{0}$. Let $\hat{Q}_{L,r}$ and $\hat{Q}_{R,r}$ be the fixed estimated left and right bases. They are orthogonal matrices. Rotating into these bases gives the matrix proposal on which normalization or sign acts:

$$\mathbf{Z}_r = \hat{Q}_{L,r}^\top M_r \hat{Q}_{R,r}, \quad \mathbf{Q}_r = \hat{Q}_{R,r} \otimes \hat{Q}_{L,r}, \quad \text{vec}(\mathbf{Z}_r) = \mathbf{Q}_r^\top \mathbf{x}_r. \quad (\text{A.9})$$

No Kronecker assumption is imposed on Σ_r here: the Kronecker matrix $\mathbf{Q}_r$ represents the two-sided rotation, not a factorization of the full covariance.

To distinguish scale error from an imperfect basis, let Γ_r be the true covariance in the estimated basis, D_r its diagonal of coordinate variances, and $\mathcal{C}_r$ the correlation matrix remaining after exact variance normalization:

$$\Gamma_r = \mathbf{Q}_r^\top \Sigma_r \mathbf{Q}_r, \quad D_r = \text{Diag}(\mathbf{v}_r), \quad v_{rj} = (\Gamma_r)_{jj}, \quad \mathcal{C}_r = D_r^{-1/2} \Gamma_r D_r^{-1/2}. \quad (\text{A.10})$$

Thus $\mathcal{C}_r$ has unit diagonal; its off-diagonal entries measure correlations that the basis has not removed. The actual normalization uses frozen variance estimates $\hat{v}_{rj}$. Write their relative errors as e_{rj} and collect the resulting inverse-scale gains in $\mathbf{A}_r$:

$$\hat{v}_{rj} = v_{rj}(1 + e_{rj}) > 0, \quad \hat{D}_r = \text{Diag}(\hat{\mathbf{v}}_r), \quad \mathbf{A}_r = \text{Diag}((1 + e_{rj})^{-1/2}). \quad (\text{A.11})$$

If $\hat{\mathbf{V}}_r$ is the $m \times n$ reshape of $\hat{\mathbf{v}}_r$, the SOAP core divides by these estimated scales, whereas SPlus takes signs. Both then rotate the result back to the parameter coordinates:

$$\begin{aligned} \mathbf{U}_r^{\text{SOAP}} &= \hat{Q}_{L,r} \left(\mathbf{Z}_r \oslash \sqrt{\hat{\mathbf{V}}_r} \right) \hat{Q}_{R,r}^\top, \\ \text{vec}(\mathbf{U}_r^{\text{SOAP}}) &= \mathbf{Q}_r \hat{D}_r^{-1/2} \mathbf{Q}_r^\top \mathbf{x}_r, \end{aligned} \quad (\text{A.12})$$

$$\begin{aligned} \mathbf{U}_r^{\text{SPlus}} &= \hat{Q}_{L,r} \text{sign}(\mathbf{Z}_r) \hat{Q}_{R,r}^\top, \\ \text{vec}(\mathbf{U}_r^{\text{SPlus}}) &= \mathbf{Q}_r \text{sign}(\mathbf{Q}_r^\top \mathbf{x}_r). \end{aligned} \quad (\text{A.13})$$

Division, square roots, and signs in these matrix expressions act entrywise.

Proposition 5 (Separate effects of the basis and coordinate scales). *Under the preceding frozen-state assumptions, define $\mathbf{C}_r^{\text{SOAP}} = \text{Cov}(\text{vec}(\mathbf{U}_r^{\text{SOAP}}))$. For every centered proposal with finite covariance,*

$$\mathbf{C}_r^{\text{SOAP}} = \mathbf{Q}_r \mathbf{A}_r \mathcal{C}_r \mathbf{A}_r^\top \mathbf{Q}_r^\top, \quad (\text{A.14})$$

and the whitening error decomposes exactly as

$$\|\mathbf{C}_r^{\text{SOAP}} - \mathbf{I}\|_F^2 = \underbrace{\sum_j \left(\frac{e_{rj}}{1 + e_{rj}} \right)^2}_{\text{coordinate-scale error}} + \underbrace{\sum_{i \neq j} \frac{\mathcal{C}_{r,ij}^2}{(1 + e_{ri})(1 + e_{rj})}}_{\text{residual correlations after normalization}}. \quad (\text{A.15})$$

If additionally $\mathbf{x}_r$ is centered Gaussian, then

$$\mathbf{C}_r^{\text{SPlus}} = \mathbf{Q}_r \left[\frac{2}{\pi} \arcsin(\mathbf{C}_r) \right] \mathbf{Q}_r^\top, \quad (\text{A.16})$$

$$\|\mathbf{C}_r^{\text{SPlus}} - \mathbf{I}\|_F^2 = \sum_{i \neq j} \left[\frac{2}{\pi} \arcsin(C_{r,ij}) \right]^2 \leq \|\mathbf{C}_r - \mathbf{I}\|_F^2. \quad (\text{A.17})$$

Here $\arcsin$ acts entrywise. The same sign identities hold for the positive scale-mixture model $\mathbf{x}_r = \sqrt{\tau_r} \mathbf{y}_r$, where $\mathbf{y}_r \sim \mathcal{N}(\mathbf{0}, \Sigma_r)$, $\tau_r > 0$ is independent of $\mathbf{y}_r$, and $\mathbb{E}[\tau_r] = 1$.

Proof. Vectorization gives Equations (A.12) and (A.13). Since the maps and denominators are frozen, the SOAP covariance is the linear covariance sandwich. Substitute $\Gamma_r = \mathbf{D}_r^{1/2} \mathbf{C}_r \mathbf{D}_r^{1/2}$ and $\hat{\mathbf{D}}_r^{-1/2} \mathbf{D}_r^{1/2} = \mathbf{A}_r$ to obtain Equation (A.14). Orthogonal conjugation preserves Frobenius norm. In the rotated frame, the diagonal covariance entries are $(1 + e_{rj})^{-1}$ and the off-diagonal entries are $C_{r,ij} / \sqrt{(1 + e_{ri})(1 + e_{rj})}$. Summing their squared differences from the identity proves Equation (A.15).

For the sign map, each centered nondegenerate Gaussian coordinate has an equiprobable sign and hence mean zero and variance one. Two standardized Gaussian coordinates with correlation ρ have sign disagreement probability $\arccos(\rho)/\pi$: represent them as projections of an isotropic two-dimensional Gaussian onto two unit directions with angle $\arccos(\rho)$ and use its uniform angular distribution. Their sign product therefore has expectation $1 - 2 \arccos(\rho)/\pi = (2/\pi) \arcsin(\rho)$. This is the Gaussian arcsine identity used in one-bit covariance estimation (Dirksen et al., 2022), and proves Equation (A.16). For $0 \leq t \leq 1$, convexity of $\arcsin(t)$ places it below the chord joining its endpoint values, so $(2/\pi) \arcsin(t) \leq t$. Oddness extends the absolute-value inequality to $[-1, 1]$. Squaring the off-diagonal entries proves Equation (A.17). Finally, multiplication by the positive scalar $\sqrt{\tau_r}$ leaves every sign unchanged; independence and $\mathbb{E}[\tau_r] = 1$ preserve the stated input covariance. $\square$

An accurate basis does not remove SOAP’s scale-estimation error. The first term remains when the basis decorrelates the proposal but the local scales are inaccurate; the second remains when exact scales are used in a basis that leaves correlations. SPlus removes the estimated-scale term in this centered model, while retaining dependence on those correlations. In particular, if $\mathbf{C}_r = \mathbf{I}$, SPlus has identity covariance for every worker, including workers using different bases and different input variances.

For an explicit bound, if $\max_j |e_{rj}| \leq \epsilon_r < 1$, then

$$\|\mathbf{C}_r^{\text{SOAP}} - \mathbf{I}\|_F^2 \leq \frac{\|\mathbf{e}_r\|_2^2 + \|\mathbf{C}_r - \mathbf{I}\|_F^2}{(1 - \epsilon_r)^2}. \quad (\text{A.18})$$

This follows by bounding each denominator in Equation (A.15) from below. Underestimating a variance can therefore amplify the whitening error through its reciprocal. Rotation back, by contrast, preserves the error in either operator or Frobenius norm:

$$\|\mathbf{Q}_r \mathbf{B} \mathbf{Q}_r^\top - \mathbf{I}\| = \|\mathbf{B} - \mathbf{I}\|. \quad (\text{A.19})$$

Orthogonal multiplication preserves singular values, proving Equation (A.19). Rotation back can change the directions of residual anisotropy, but cannot increase this norm of its error.

B TRAINING SETUP AND HYPERPARAMETERS

Models and data. The experiments use decoder-only LLaMA models trained on DCLM (Li et al., 2024), with sequence length 2048 and evaluation on a held-out 100M-token validation split. The model configuration files give the architecture below. Global batch size is held fixed within each model scale, so the local batch is the global batch divided by R .

Setting	178M	512M
Transformer layers	9	12
Hidden width	1024	1536
Attention heads / key-value heads	8/8	12/12
Feed-forward width	2688	5440
Activation	SwiGLU	SwiGLU
Sequence length	2048	2048
Global tokens per inner step	524,288	2,097,152
Global sequences per inner step	256	1024
Training tokens (approximately)	3.55B	10.25B

Table 7: Architecture and training scales. The global batch is measured per inner step, not per communication round.

Aggregation and comparisons. Unless stated otherwise, replicas take $H = 30$ local steps and use a Nesterov SGD outer optimizer. The $H = 1$ controls use uniform averaging without outer momentum. Percentage changes in validation loss use each optimizer’s own DP reference and are computed before rounding. The covariance, mean/noise, and curvature probes use the 178M outer-step-75 checkpoints; their replay settings and sample counts are specified in Appendix C.

Hyperparameter sweep. We tuned the learning rate for each optimizer in the DP setting and the inner and outer learning rates and outer momentum for local training. For each optimizer we tuned $R=8,64,96$ each with at least 10 grid points extrapolating the HP for $R=16$ and 32.

C ANALYSIS PROTOCOLS AND ESTIMATORS

All three analyses use the 178M outer-step-75 checkpoints. The covariance and curvature probes use six matrices (attention query and MLP down-projection at layers 0, 4, and 8); the DP-referenced mean/noise comparison covers all 63 hidden attention and MLP matrices. Each optimizer uses its own trained checkpoint. The protocols below distinguish native saved states from the common-state replay used for comparison with DP.

C.1 CONDITIONAL COVARIANCE REPLAY

This appendix spells out the covariance measurement used in Table 4. The probe estimates the minibatch-induced distribution of an optimizer’s local update conditional on one trained checkpoint. It does not treat successive updates from the training trajectory as independent covariance samples.

C.1.1 CONDITIONAL REPLAY PROTOCOL

At a communication boundary, all replicas share the synchronized parameters $\mathbf{W}$, while replica r has its own saved optimizer state s_r . Table 4 uses all 64 saved replica states. For each replica, we run $2n$ rollouts: $n = 16$. Before every rollout, we restore the same pair $(\mathbf{W}, s_r)$, then consume a new segment of the replica’s shuffled data stream. Within a rollout, the parameters and optimizer state evolve normally. We record the parameter displacement after $h \in \{1, 30\}$ local steps,

$$\mathbf{u}_{r,j}^{(h)} = \text{concat}_{\ell \in \mathcal{L}} \text{vec} \left(\mathbf{W}_\ell - \mathbf{W}_{r,j,\ell}^{(h)} \right), \quad j = 1, \dots, 2n, \quad (\text{C.1})$$

where $\mathcal{L}$ contains attention `q_proj` and MLP `down_proj` at layers 0, 4, and 8. For the 178M model, this concatenates $d = 11,403,264$ coordinates. The $h = 1$ and $h = 30$ endpoints come from the same rollout and therefore have coupled finite-sample errors.

The per-replica batch is four sequences for $R = 64$, giving a global batch of 256 sequences. Each replay uses the optimizer configuration saved by the corresponding training run, including its replica-local momentum and second moments and its saved replica-specific or shared basis.

For fixed r and h , write

$$\boldsymbol{\mu}_{r,h} = \mathbb{E}[\mathbf{u}_{r,j}^{(h)}], \quad \mathbf{C}_{r,h} = \text{Cov}(\mathbf{u}_{r,j}^{(h)}). \quad (\text{C.2})$$

Two consecutive rollouts form one paired difference,

$$\mathbf{z}_{r,i}^{(h)} = \frac{\mathbf{u}_{r,2i-1}^{(h)} - \mathbf{u}_{r,2i}^{(h)}}{\sqrt{2}}, \quad i = 1, \dots, n. \quad (\text{C.3})$$

If the two rollouts are independent draws from the same conditional distribution, then

$$\mathbb{E}[\mathbf{z}_{r,i}^{(h)}] = \mathbf{0}, \quad \text{Cov}(\mathbf{z}_{r,i}^{(h)}) = \frac{1}{2}(\mathbf{C}_{r,h} + \mathbf{C}_{r,h}) = \mathbf{C}_{r,h}. \quad (\text{C.4})$$

The subtraction therefore removes the unknown mean, while $1/\sqrt{2}$ restores the original covariance scale.

C.1.2 TRACE ESTIMATORS FROM FULL-COORDINATE INNER PRODUCTS

Fix a horizon and suppress h in the notation. Define the Gram entries

$$G_{ri,sj} = \mathbf{z}_{r,i}^\top \mathbf{z}_{s,j}. \quad (\text{C.5})$$

The code computes these inner products over every selected coordinate; it does not use a random projection. It nevertheless estimates population moments from n paired differences per replica.

For each replica, define

$$t_r = \frac{1}{n} \sum_{i=1}^n G_{ri,ri}, \quad (\text{C.6})$$

$$b_r = \frac{1}{n(n-1)} \sum_{i \neq j} G_{ri,ri} G_{rj,rj}, \quad (\text{C.7})$$

$$q_{rr} = \frac{1}{n(n-1)} \sum_{i \neq j} G_{ri,rj}^2. \quad (\text{C.8})$$

Under the IID-rollout idealization, for independent paired differences,

$$\mathbb{E}[t_r] = \text{tr}(\mathbf{C}_r), \quad \mathbb{E}[b_r] = \text{tr}(\mathbf{C}_r)^2, \quad \mathbb{E}[q_{rr}] = \text{tr}(\mathbf{C}_r^2). \quad (\text{C.9})$$

The last identity follows from

$$\mathbb{E}[(\mathbf{z}^\top \mathbf{z}')^2] = \text{tr}(\mathbb{E}[\mathbf{z} \mathbf{z}^\top] \mathbb{E}[\mathbf{z}' \mathbf{z}'^\top]) = \text{tr}(\mathbf{C}^2) \quad (\text{C.10})$$

for independent $\mathbf{z}$ and $\mathbf{z}'$. Using $i \neq j$ is important: including the diagonal would add fourth-moment self-products and bias the estimate upward.

For different replicas $r \neq s$, the cross term

$$q_{rs} = \frac{1}{n^2} \sum_{i,j} G_{ri,sj}^2 \quad (\text{C.11})$$

estimates $\text{tr}(\mathbf{C}_r \mathbf{C}_s)$ under the same IID and conditional replica-independence idealization. The quantity is the scale-dependent Frobenius overlap between two replicas' covariance matrices, not a cross-covariance between their stochastic updates.

C.1.3 SCALED SUM OF LOCAL COVARIANCES

The probe constructs the scaled sum

$$\tilde{\mathbf{C}} = \frac{1}{R^2} \sum_{r=1}^R \mathbf{C}_r. \quad (\text{C.12})$$

For the actual replica average $\bar{\mathbf{u}} = R^{-1} \sum_r \mathbf{u}_r$, this equals $\text{Cov}(\bar{\mathbf{u}})$ if the cross-replica noise covariances vanish conditional on the starting parameters and optimizer states. The local-replay probe

does not measure those cross-covariances. To estimate the trace moments of $\tilde{\mathbf{C}}$, it uses

$$\hat{T} = \frac{1}{R^2} \sum_r t_r, \quad \mathbb{E}[\hat{T}] = \text{tr}(\tilde{\mathbf{C}}), \quad (\text{C.13})$$

$$\hat{S} = \frac{1}{R^4} \sum_{r,s} q_{rs}, \quad \mathbb{E}[\hat{S}] = \text{tr}(\tilde{\mathbf{C}}^2), \quad (\text{C.14})$$

$$\hat{B} = \frac{1}{R^4} \left(\sum_r b_r + \sum_{r \neq s} t_r t_s \right), \quad \mathbb{E}[\hat{B}] = \text{tr}(\tilde{\mathbf{C}}^2). \quad (\text{C.15})$$

The separate estimator $\hat{B}$ avoids squaring the noisy estimate $\hat{T}$. The reported anisotropy is

$$\hat{a} = \sqrt{\max \left\{ 0, \frac{d\hat{S}}{\hat{B}} - 1 \right\}}. \quad (\text{C.16})$$

Under the IID idealization, $\hat{T}$, $\hat{S}$, and $\hat{B}$ are built from unbiased moment estimators, but their ratio and square root are not themselves unbiased. The clipping only handles a negative raw value caused by finite-sample error; it can therefore return zero even when the population covariance is anisotropic.

If $\lambda_1, \dots, \lambda_d$ are the eigenvalues of $\tilde{\mathbf{C}}$, the population quantity is

$$a^2 = \frac{d \sum_i \lambda_i^2}{(\sum_i \lambda_i)^2} - 1, \quad r_{\text{eff}} = \frac{1}{1 + a^2}, \quad (\text{C.17})$$

where $r_{\text{eff}} = (\sum_i \lambda_i)^2 / \sum_i \lambda_i^2$ is the participation-ratio effective rank. Thus a measures covariance shape independently of scale; it is undefined when the covariance is identically zero. For example, $\text{diag}(5, 5)$ has $a = 0$, whereas $\text{diag}(9, 1)$ has $a = 0.8$. Multiplying either matrix by a positive scalar leaves a unchanged.

C.1.4 SCOPE AND INTERPRETATION

The estimator is conditional on one checkpoint and one set of replica-local states. Each method is evaluated on its own trained trajectory, so a cross-method difference can reflect both the update rule and the state reached by training; it is not a same-state causal ablation. Concatenating six matrices also mixes within-matrix anisotropy, scale imbalance between matrices, and cross-matrix covariance. The input streams are disjoint segments of one shuffled replica-local permutation, rather than independent samples drawn with replacement. The unbiasedness equalities above are therefore exact only under the stated IID idealization. Moreover, the $n(n-1)$ and n^2 sums reuse the same n vectors; their terms are not independent observations. Finally, interpreting Equation (C.12) as the covariance of the actual average requires vanishing conditional cross-replica noise covariances.

Lower a indicates a more isotropic $\tilde{\mathbf{C}}$, independently of total noise magnitude. For SOAP, the saved position relative to a basis-refresh step can also affect the measured covariance: the fresh gradient may enter both the basis and the proposal. At this single checkpoint, the probe characterizes update geometry but does not identify its causal contribution to final loss.

C.2 DP-REFERENCED DIRECTIONAL MEAN AND NOISE

Each optimizer is evaluated at its own round-75 checkpoint. At the start of every repetition, the checkpoint weights and replica 0's saved optimizer state are copied to every replica and to the DP reference. "Common state" means identical states within that replay, not an averaged optimizer state or a state shared across optimizers. Each replica consumes a persistent, shuffled data stream without reusing sequences across repetitions. In each step DP sees the union of the same local batches, evaluated at its own iterate. This preserves the pairing between local and DP data without reusing examples between the direction-fitting and evaluation samples.

Let j index one of the 63 hidden matrices, vectorized in parameter coordinates. Write $\mathbf{x}_{j,k}$ for its averaged local displacement and $\mathbf{y}_{j,k}$ for its DP displacement in repetition k . On the 16 fitting repetitions at $R = 8$, define

$$\hat{\alpha}_j = \frac{\langle \bar{\mathbf{x}}_{j,\text{fit}}, \bar{\mathbf{y}}_{j,\text{fit}} \rangle}{\|\bar{\mathbf{x}}_{j,\text{fit}}\|^2},$$

and take $\hat{\mathbf{q}}_j$ to be the leading eigenvector of the sample covariance of $\hat{\alpha}_j \mathbf{x}_{j,k}$ on those same fitting repetitions. The reference mean is $\mathbf{d}_j = \mathbb{E}[\mathbf{y}_{j,k}]$, and the mean discrepancy is $\mathbf{b}_j = \hat{\alpha}_j \mathbb{E}[\mathbf{x}_{j,k}] - \mathbf{d}_j$, conditional on the fitting sample. The decomposition in Equation (10) therefore applies to each matrix with these fitted quantities held fixed.

We use the remaining $K = 24$ independent repetitions for evaluation. Set $\mathbf{z}_{j,k} = \hat{\alpha}_j \mathbf{x}_{j,k} - \mathbf{y}_{j,k}$, $t_{j,k} = \hat{\mathbf{q}}_j^\top \mathbf{z}_{j,k}$, and $p_{j,k} = \hat{\alpha}_j \hat{\mathbf{q}}_j^\top \mathbf{x}_{j,k}$. With s^2 denoting the unbiased sample variance on these repetitions, the unnormalized estimates are

$$\hat{B}_j = \bar{t}_j^2 - \frac{s^2(t_{j,k})}{K}, \quad \hat{N}_j = s^2(p_{j,k}).$$

Conditional on the fit and independent evaluation repetitions, $\hat{B}_j$ is unbiased for $(\hat{\mathbf{q}}_j^\top \mathbf{b}_j)^2$ and $\hat{N}_j$ estimates $\hat{\alpha}_j^2 \hat{\mathbf{q}}_j^\top \mathbf{C}_j \hat{\mathbf{q}}_j$, where $\mathbf{C}_j = \text{Cov}(\mathbf{x}_{j,k})$. Crucially, the correction uses the variance of the *paired* difference $\mathbf{z}_{j,k}$; it retains the local-DP cross-covariance. The stochastic term is the variance of a single aggregated update, not that variance divided by the number of evaluation repetitions.

For each matrix, the first two numerical columns in Table 5 are $\hat{B}_j / \|\hat{\mathbf{d}}_j\|^2$ and $\hat{N}_j / \|\hat{\mathbf{d}}_j\|^2$, with $\hat{\mathbf{d}}_j = K^{-1} \sum_k \mathbf{y}_{j,k}$, expressed in units of 10^{-6} . The third column is $100 \hat{B}_j / \hat{N}_j$. Every column is averaged with equal weight over the 63 matrices. In particular, the reported average ratio is not the ratio of the two reported averages. Bias-corrected mismatch estimates can be negative and are not clipped before averaging. The estimated DP normalizer and the ratios introduce additional finite-sample uncertainty; the normalized entries are not claimed to be unbiased. The matrices share data and are not independent experimental repeats.

Trace-based comparison. The corresponding whole-matrix estimates are

$$\hat{B}_j^{\text{tr}} = \|\bar{\mathbf{z}}_j\|^2 - \frac{1}{K(K-1)} \sum_k \|\mathbf{z}_{j,k} - \bar{\mathbf{z}}_j\|^2, \quad \hat{N}_j^{\text{tr}} = \frac{\hat{\alpha}_j^2}{K-1} \sum_k \|\mathbf{x}_{j,k} - \bar{\mathbf{x}}_j\|^2.$$

They estimate $\|\mathbf{b}_j\|^2$ and $\hat{\alpha}_j^2 \text{tr}(\mathbf{C}_j)$, respectively. The same DP-mean normalizer and equal per-matrix averaging apply; the mean/noise ratio is computed per matrix before averaging. At $R = 8, H = 30$, these average ratios are 6.2% for SPlus, 2.8% for SOAP, 10% for Muon, and 6.7% for AdamW.

C.3 CURVATURE ALONG FITTED HIGH-NOISE DIRECTIONS

At each method’s own checkpoint, restore the saved parameters and optimizer state before every one-step replay. Form 128 paired update differences $\mathbf{z}_i = (\mathbf{u}_i - \mathbf{u}'_i)/\sqrt{2}$, using fresh local batches of $b = 256/R$ sequences. Under the independent-replay model these differences are centered and have the local-update covariance. For Table 6, use the first 64 of the 128 paired samples to select

$$\hat{\mathbf{q}} = \arg \max_{\|\mathbf{q}\|=1} \frac{1}{64} \sum_{i=1}^{64} (\mathbf{q}^\top \mathbf{z}_i)^2.$$

Hessian–vector products on 64 held-out validation sequences evaluate curvature at the same checkpoint parameters. Draw 16 independent standard Gaussian vectors in the six-matrix subspace and normalize each to unit length, giving $\mathbf{v}_1, \dots, \mathbf{v}_{16}$. Report

$$\kappa(\hat{\mathbf{q}}) = \frac{\hat{\mathbf{q}}^\top \mathcal{H} \hat{\mathbf{q}}}{\frac{1}{16} \sum_{j=1}^{16} \mathbf{v}_j^\top \mathcal{H} \mathbf{v}_j}.$$

The numerator measures curvature along a direction selected for high noise variance, not maximum curvature. The denominator estimates $\text{tr}(\mathcal{H})/d$ in this subspace. Each method uses its own trained checkpoint, so the Hessian and reference curvature are method-specific. Ranges span the two estimates from replicas 0 and 1; a single displayed value means both endpoints round to it. These are one-step local-update measurements, not $H = 30$ aggregate measurements. The ratio removes the noise magnitude and does not estimate $\text{tr}(\mathcal{H}\mathbf{C})$.