LISC: BALANCING SAMPLE INFLUENCE WITH SPECTRAL REWEIGHTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern datasets are highly heterogeneous, with predictive signals unevenly distributed across samples. As a consequence, gradients estimated as averages over mini-batches tend to reinforce dominant learning signals, leaving other potentially useful directions weakly represented. This issue cannot be directly addressed by optimizers that operate strictly on gradients already aggregated across samples. We therefore propose a sample-reweighting method, Local Inverse Spectral Curriculum (LISC), which re-balances gradient directions while remaining fully compatible with existing optimizers. Starting from a local linear approximation of per-example losses, LISC assigns weights that reduce the relative influence of strongly represented gradient directions. Its underlying spectral transformation is equivalent to preconditioning with a local gradient second moment, yet it avoids constructing a full parameter-space preconditioner. Across vision and language tasks, architectures, and optimizers, LISC improves early-training test performance, with gains exceeding five percentage points in multiple settings, while matching final performance in vision and improving final generalization in several language tasks. These findings indicate that geometry-aware sample aggregation can complement optimizer-level transformations to accelerate learning without sacrificing, and in some settings even improving, final generalization.

1 INTRODUCTION

Mini-batch training involves two distinct operations: examples are first aggregated into a batch gradient, and an optimizer then transforms that aggregate into a parameter update. Most optimization methods focus on the second operation. Yet, the first is equally consequential. When many examples induce similar gradients, their shared directions are reinforced, while complementary directions represented by less-aligned examples can contribute much less to the resulting update (Yin et al., 2018; Sankararaman et al., 2020). Thus, equal sample weights need not imply equal influence across gradient directions.

This distinction matters because the optimizer only acts *after* aggregation. Methods such as AdamW, Shampoo, or Muon (Loshchilov & Hutter, 2019; Gupta et al., 2018; Jordan et al., 2024) can rescale, precondition, or otherwise transform the batch gradient they receive, but they do not directly control which examples determined that gradient in the first place. Once per-example gradients have been compressed into a single aggregate, information about their redundancy and interactions is no longer explicitly available to the optimizer. Consequently, a potentially important degree of freedom in training is left largely untouched: how should examples be aggregated before the optimizer acts?

This question is particularly relevant for heterogeneous data, where examples can carry substantially different predictive signals (Koh et al., 2021; Sagawa et al., 2020; Swayamdipta et al., 2020), and changing their relative influence can materially affect learning (Xie et al., 2023). Figure 1 shows that gradient energy can be highly concentrated in the leading spectral mode at initialization, with substantially stronger concentration in the language settings that we consider. Neural networks can also acquire different predictive structures at markedly different rates (Arpit et al., 2017; Rahaman et al., 2019; Shah et al., 2020; Pezeshki et al., 2021). This becomes particularly important under finite optimization budgets, where large datasets and compute constraints often make training to convergence impractical (Kaplan et al., 2020; Hoffmann et al., 2022). In this regime, not only what the model eventually learns, but also how quickly it acquires useful predictive structure, becomes

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

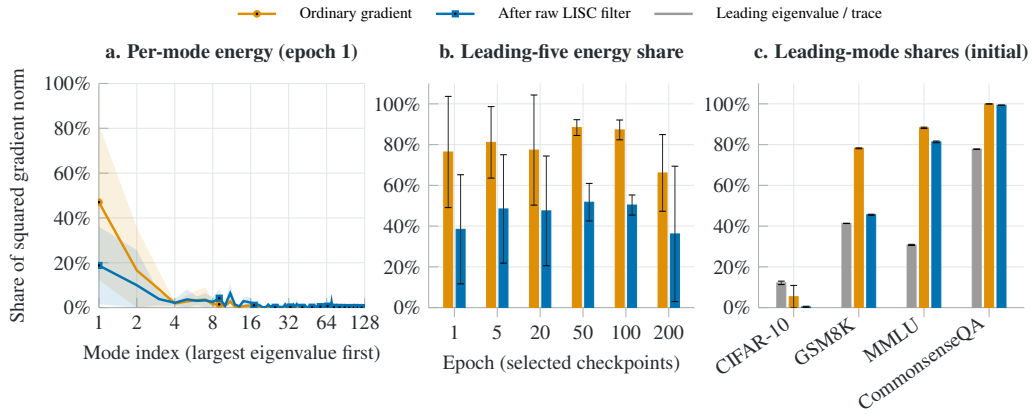


Figure 1: **Gradient energy concentrates in leading spectral modes; raw LISC redistributes it across modes.** **(a)** Per-mode gradient energy at epoch 1 for ResNet-18 on CIFAR-10. **(b)** Fraction of gradient energy captured by the top five modes over training. At epoch 1, LISC reduces this fraction from 76.4% to 38.4%. **(c)** Energy in the leading mode at initialization, revealing substantially stronger spectral concentration in language tasks and motivating more aggressive spectral rebalancing.

important. This motivates the central question of this work: *Can learning be improved by controlling the geometry of sample aggregation before the optimizer acts?*

Several approaches have been proposed to address aspects related to this problem. Curriculum learning and sample reweighting adapt example importance using scalar scores (Bengio et al., 2009; Ren et al., 2018; Paul et al., 2021), but such scores do not directly capture redundancy between gradient directions. Conversely, gradient-balancing and preconditioning methods exploit optimization geometry (Gupta et al., 2018; Senushkin et al., 2023), but typically modify parameter updates rather than constructing a sample-weighting curriculum. We bridge these views by using mini-batch gradient geometry to determine how examples should be weighted.

We introduce **Local Inverse Spectral Curriculum (LISC)**, a sample-reweighting method motivated by a local linear approximation of per-example losses. This approximation links sample weights to local loss changes through a sample-gradient Gram matrix. Its spectrum reveals directions that are repeatedly represented across the batch. LISC applies a damped inverse spectral filter to this matrix, attenuating strongly represented modes relative to weaker complementary ones, and converts the resulting coefficients into sample weights. The geometry is recomputed as the model evolves, yielding a curriculum that is local to both the mini-batch and the model state.

The raw LISC transformation is equivalent to applying an inverse-square-root preconditioner based on the local gradient second moment, connecting sample reweighting to structured preconditioning methods (Gupta et al., 2018; Vyas et al., 2025). This equivalence does not require constructing a parameter-space preconditioning matrix. The transformation can instead be realized through the $b \times b$ sample-gradient Gram matrix. Because LISC changes the gradient before it reaches the optimizer, it is complementary to optimizer design and can be combined with different optimization methods.

An efficient approximation makes LISC practical for both standard training and parameter-efficient fine-tuning (Hu et al., 2022). We evaluate it across vision and language tasks, architectures, and optimizers and find that LISC consistently improves learning per optimization step across diverse settings. On CIFAR-100 with ResNet-18 and SGD, for example, epoch-five accuracy improves from 42.68% to 47.70%, while final accuracy increases from 78.26% to 78.97%. Notably, these gains can occur even when LISC has higher ordinary training loss, suggesting that its benefit is not simply faster empirical-risk minimization, but a change in the order and relative rate at which predictive structure is acquired.

Contributions:

- **Directional concentration.** We show that gradient energy concentrates in leading spectral modes, with substantially stronger concentration in the language settings that we evaluate. This poses a challenge for learning underrepresented samples.

- **Geometry-aware sample aggregation.** We introduce LISC to rebalance gradient directions before the optimizer acts and prove, in a rank-one construction, that it enables directional changes unavailable to Muon alone, yielding strictly greater first-order test-loss improvement.
- **A preconditioning connection.** We prove that raw LISC is equivalent to local inverse-square-root second-moment preconditioning and characterize its spectral and descent properties.
- **Efficient implementation and empirical gains.** We provide an efficient implementation based on an exact Gram-matrix factorization and evaluate LISC across vision and language settings, demonstrating improved early test performance and competitive or improved final generalization.

2 Related Work

Sample reweighting and data selection. Curriculum and self-paced learning adjust example importance using difficulty (Bengio et al., 2009; Kumar et al., 2010), while meta-reweighting learns weights from validation signals (Ren et al., 2018; Shu et al., 2019). Other approaches use pretrained-model scores (Sanyal et al., 2025), gradient norms, or prediction errors (Paul et al., 2021). Gradient matching selects weighted subsets that approximate a target gradient (Killamsetty et al., 2021), and InfoBatch prunes examples while preserving the expected update (Qin et al., 2024). LISC instead derives weights jointly from within-batch gradient interactions to deliberately redistribute the aggregate gradient's spectral energy.

Multi-objective gradient aggregation. PCGrad projects task gradients to mitigate conflict (Yu et al., 2020), while Aligned-MTL applies an inverse-square-root task-Gram transformation to equalize the nonzero singular values of the gradient matrix (Senushkin et al., 2023). Jacobian descent provides a broader framework for aggregating objective gradients, including individual-example losses, and evaluates Aligned-MTL in this setting (Quinton & Rey, 2024). Its proposed UPGrad aggregator targets updates that do not conflict with any objective. LISC instead reduces spectral concentration without requiring every example's loss to decrease. Crucially, weakly represented modes are not necessarily useful, and dominant modes are not necessarily undesirable. Complete spectral equalization therefore need not be the appropriate objective. Finite damping allows LISC to temper spectral dominance while retaining differences in spectral strength that may encode useful structure. We also provide an efficient implementation using selected measurement coordinates.

Related geometric reweighting. SALT uses sample-gradient Gram geometry, but in a different context. It addresses signed cancellation in group-relative policy optimization by separating shared and residual subspaces and amplifying the residual channel (Chang et al., 2026). In contrast, LISC applies a smooth inverse spectral filter across modes and constructs nonnegative weights for supervised losses. EigenPro and its scalable variants accelerate kernel learning through spectral preconditioning (Ma & Belkin, 2017; Abedsoltan et al., 2025). LISC instead recomputes its operator from the current mini-batch's loss gradients as the model evolves.

Optimizer-level preconditioning. Shampoo and SOAP use structured gradient moments (Gupta et al., 2018; Vyas et al., 2025), while SWAN and Muon whiten gradient matrices or orthogonalize momentum updates (Ma et al., 2025; Jordan et al., 2024; Liu et al., 2025). Faster learning can also be achieved by implicit bias control (Gadhikar et al., 2025; Jacobs et al., 2026a;b) and rescaling parameters to balance gradients (Mustafa & Burkholz, 2024). These methods transform parameter updates, while LISC determines sample weights before aggregation and can therefore be combined with them. Before clipping and normalization, its update in the measurement coordinates is exactly equivalent to damped inverse-square-root preconditioning by the local gradient second moment. This equivalence requires neither an explicit parameter-space preconditioner nor identifying gradient second moments with curvature (Martens, 2020; Kunstner et al., 2019).

3 Preliminaries

Consider a mini-batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{b}$ with per-example losses $\ell_i(\theta)$ and reference weights $m \in \Delta_b$. For standard classification, $m = \mathbf{1}/b$; for language-modeling, if $\ell_{t,i}$ is the mean loss over n_i tokens in sequence i, we use $m_{t,i} = n_i / \sum_j n_j$ to recover the batch token-averaged objective.

We distinguish the trainable parameters $\theta \in \mathbb{R}^d$ from the measurement coordinates $\phi \in \mathbb{R}^q$ used to measure interactions among examples, with $q \leq d$. We take ϕ to be either all trainable parameters or a selected subset of coordinates of θ, such as the classifier. At the current parameter state, define

$$a_{t,i} = \nabla_\phi \ell_{t,i}(\theta_t) \in \mathbb{R}^q, \qquad A_t = \begin{bmatrix} a_{t,1}^\top \\ \vdots \\ a_{t,b}^\top \end{bmatrix} \in \mathbb{R}^{b \times q}, \qquad K_t = A_t A_t^\top \in \mathbb{R}^{b \times b}.$$

Here, K_t is the sample-gradient Gram matrix, with entries $(K_t)_{ij} = \langle a_{t,i}, a_{t,j} \rangle$. Its diagonal entries record squared per-example gradient norms, whereas its off-diagonal entries capture gradient agreement between examples. For simplicity, we suppress the iteration index t.

Lemma 1 (Local effect of sample weighting)**.** *Under a local linearization of the per-example losses with respect to ϕ, for fixed $w \in \mathbb{R}^b$ and $g_w = A^\top w$,*

$$\boldsymbol{\ell}(\phi - \eta g_w) = \boldsymbol{\ell}(\phi) - \eta K w + o(\eta).$$

Consequently, for $g := g_m = A^\top m$, the first-order change in the reference objective is

$$F_t(\phi - \eta g_w) - F_t(\phi) = -\eta \langle g, g_w \rangle + o(\eta) = -\eta m^\top K w + o(\eta).$$

Lemma 1 links sample weighting, parameter-space updates, and local changes in the individual losses through the same Gram operator K (Quinton & Rey, 2024, Sec. 2.1). Importantly, even nonnegative sample weights do not guarantee that every example's loss decreases, since individual entries of Kw can be negative. The proof is provided in Appendix A.2.

Let the compact SVD be $A = U \Sigma V^\top$, with $\lambda_j = \sigma_j^2$ ordered as $\lambda_1 \geq \cdots \geq \lambda_r > 0$. Writing $c_j = u_j^\top m$, the ordinary batch gradient in the measurement coordinates is

$$g = A^\top m = \sum_{j=1}^{r} \sqrt{\lambda_j} c_j v_j, \qquad \|g\|_2^2 = \sum_{j=1}^{r} \lambda_j c_j^2. \tag{1}$$

Hence equal weighting of examples does not imply equal influence across gradient directions. A mode contributes strongly only when it has both a large singular value and substantial overlap with the reference coefficient vector m. In particular, a large eigenvalue alone does not imply a large contribution to the gradient. A mode for which $c_j = 0$ is absent from g.

$A^\top A$ is a gradient second-moment operator, not in general the Hessian, the neural tangent kernel (Jacot et al., 2018), or the generalized Gauss–Newton (Martens, 2020). The gradient second moments should not generally be identified with curvature matrices (Kunstner et al., 2019). We therefore formulate LISC directly in terms of gradient geometry without relying on such equivalences.

4 Local Inverse Spectral Curriculum

LISC dynamically reweights examples using the local geometry of the current mini-batch. Its goal is to reduce the relative influence of gradient directions that are already strongly represented while retaining weaker complementary directions.

4.1 Inverse spectral reweighting

For $K \neq 0$, let $\tau = \gamma \lambda_{\max}(K)$ and define $f_\tau(\lambda) = \sqrt{\frac{\tau}{\lambda + \tau}}$. If $K = 0$, we leave the reference weights unchanged, $w = m$. Otherwise, we compute the raw coefficients

$$\widehat{w} = f_\tau(K) m, \qquad \widehat{g} := g_{\widehat{w}} = A^\top \widehat{w} \tag{2}$$

Since f_τ decreases with λ, modes with larger Gram eigenvalues are attenuated more strongly. The choice $\tau = \gamma \lambda_{\max}(K)$ also makes the raw coefficients invariant to a common rescaling of all measurement gradients. Smaller γ produces stronger relative attenuation of large-eigenvalue modes. Table 6 (Appendix B.4) examines this choice empirically and motivates our default $\gamma = 0.03$, which achieves the highest mean early AUC among the tested values on CIFAR-10.

The following proposition gives an equivalent parameter-space view.

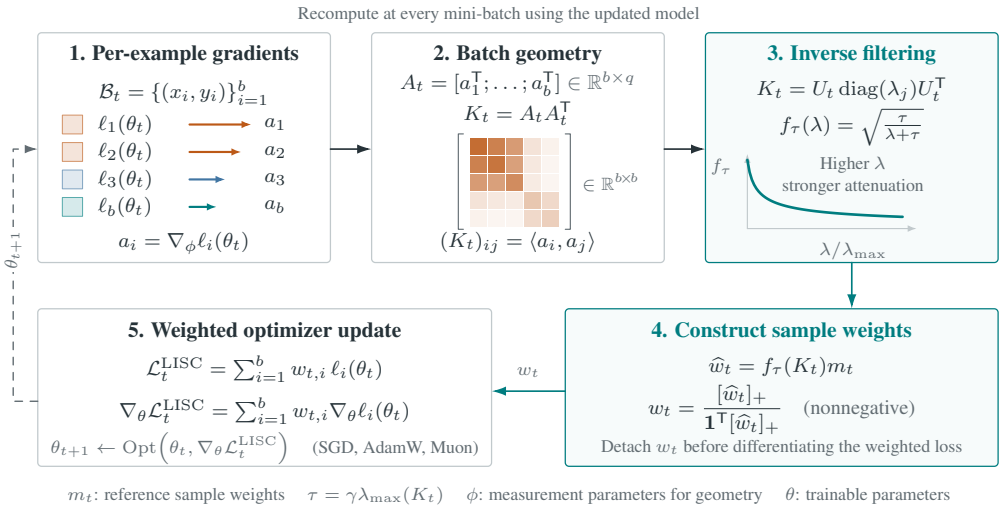


Figure 2: Overview of **Local Inverse Spectral Curriculum (LISC)**. LISC constructs a sample-gradient Gram matrix from the current mini-batch, applies a damped inverse spectral filter, and converts the resulting coefficients into nonnegative sample weights. These weights rebalance the mini-batch update across gradient directions and are recomputed at every training step.

Proposition 2 (Inverse-square-root preconditioning)**.** *Let $S = A^\top A$. Then*

$$\widehat{g} = (I + S/\tau)^{-1/2} g = \sum_{j=1}^{r} \sqrt{\lambda_j} f_\tau(\lambda_j) c_j v_j.$$

Equivalently, for any $\eta > 0$, $-\eta \widehat{g}$ uniquely minimizes $g^\top \delta + \frac{1}{2\eta} \delta^\top (I + S/\tau)^{1/2} \delta$.

Thus, LISC filtering in sample space is equivalent to metric steepest descent under $M = (I + S/\tau)^{1/2}$, or to preconditioning the reference gradient by M^{-1}, with stronger attenuation along directions of larger local gradient second moment. The damped operator $(I + S/\tau)^{-1/2}$ remains well-defined even when S is rank-deficient, avoiding the zero-curvature singularities that can make undamped inverse-Hessian preconditioning ill-defined in overparameterized models. Moreover, LISC realizes this geometry through the $\mathbb{R}^{b \times b}$ sample Gram matrix rather than an explicit $\mathbb{R}^{q \times q}$ parameter-space matrix. The proof is provided in Appendix A.3.

The same inverse-square-root response also compresses the positive spectrum of the associated raw filtered Gram operator, strictly improving its condition number whenever the positive eigenvalues are unequal (Appendix A.7). Figure 7c (Appendix B.8) illustrates this compression for classifier-head Gram matrices, comparing the condition numbers before and after raw filtering on the same retained positive modes. Under the stronger assumption that the gradient second moment matches the curvature of a quadratic objective, this response further yields a strictly better optimally tuned worst-case contraction factor than gradient descent (Appendix A.8).

4.2 Balancing Directional Influence

Let $E_j = \lambda_j c_j^2$ denote the contribution of mode j to $\|g\|_2^2$, and let $C_k = \sum_{j \le k} E_j / \sum_j E_j$ be the fraction of gradient energy contained in the k largest-eigenvalue modes. Under the LISC transformation, the corresponding energy is $\widehat{E}_j = E_j f_\tau(\lambda_j)^2$.

Theorem 3 (Suppression of leading spectral modes)**.** *For $g \neq 0$ and $\tau > 0$,*

$$\widehat{C}_k := \frac{\sum_{j \le k} \widehat{E}_j}{\sum_j \widehat{E}_j} \le C_k \qquad \text{for all } k.$$

The inequality is strict whenever energy is present on both sides of the cutoff and the corresponding eigenvalues are unequal. Furthermore, $\widehat{E}_j = c_j^2 \frac{\tau \lambda_j}{\lambda_j + \tau} \le \tau c_j^2$.

Theorem 3 shows that LISC shifts relative energy away from the leading eigenvalue modes without introducing new directions. For the inverse-square-root filter, the effective gain $\sqrt{\tau\lambda/(\lambda+\tau)}$ saturates as λ grows, compressing the spread between strongly and weakly represented modes. Figure 1(a,b) illustrates this redistribution for ResNet-18 trained on CIFAR-10: at epoch 1, raw LISC reduces the fraction of gradient energy in the five largest-eigenvalue modes from 76.4% to 38.4%. Appendix A.5 gives a sufficient condition under which such an increase in energy effective rank is guaranteed.

Proposition 4 (Alignment and Contraction)**.** *Assume $g \neq 0$ and $\tau = \gamma\lambda_1$ with $\gamma > 0$. LISC update $\widehat{g}$ is smaller in norm than the mini-batch gradient g, yet remains positively aligned with it. Formally,*

$$\langle g, \widehat{g}\rangle > 0, \qquad \|\widehat{g}\|_2 \leq \|g\|_2,$$

If, additionally, the reference objective is L-smooth in a neighborhood containing η, then for any $0 < \eta < 2/L$,

$$F_t(\phi - \eta\widehat{g}) \leq F_t(\phi) - \eta\left(1 - \frac{L\eta}{2}\right)\sqrt{\frac{\gamma}{1+\gamma}}\|g\|_2^2.$$

The proof is provided in Appendix A.6. Proposition 4 clarifies a core mechanism of LISC: while the method reduces the magnitude of the update step, it preserves guaranteed descent. The reduction in update magnitude does not imply slower learning or worse generalization. Instead, as we demonstrate empirically, LISC alters the *directional allocation* of parameter updates that accelerates the acquisition of features that generalize.

From spectral coefficients to sample weights. The raw vector $\widehat{w}$ need not be elementwise nonnegative. Negative coefficients would explicitly increase the loss of the associated examples. To avoid this, we clip and renormalize: $w = \frac{[\widehat{w}]_+}{\mathbf{1}^\top[\widehat{w}]_+}$. The denominator is positive for nonzero $m \in \Delta_b$. Clipping can change direction, and normalization changes scale; Appendix A.9 quantifies both. Empirically, Figure 7b (Appendix B.8) shows that fewer than 1% of raw coefficients are non-positive throughout the reported runs. We detach w before differentiating the weighted loss $L_{\mathrm{LISC}}(\theta) = \sum_{i=1}^{b} w_i\ell_i(\theta)$.

Measurement geometry and efficient implementation. LISC does not require full per-example parameter gradients. For classification, we use the linear classifier head as the measurement coordinates. With $z_i = Wh_i + b$, $r_i = \nabla_{z_i}\ell_i$, and augmented feature $\widetilde{h}_i = [h_i^\top, 1]^\top$, the exact head-gradient Gram entries factorize as

$$K_{ij}^{\mathrm{head}} = \langle r_i, r_j\rangle\langle \widetilde{h}_i, \widetilde{h}_j\rangle. \tag{3}$$

Thus, K^{head} can be constructed without materializing per-example parameter gradients. For LoRA fine-tuning (Hu et al., 2022), we measure gradients in the trainable adapters. When ϕ is a proper subset of θ, the resulting geometry determines only the sample weights. The weighted loss is still differentiated with respect to the full trainable parameters θ. Derivations and computational details are given in Appendix A.10. After constructing K, the only additional dense linear-algebra operation is an eigendecomposition of the $b \times b$ Gram matrix.

Compatibility with existing optimizers. LISC acts entirely through detached sample weights and therefore leaves the downstream optimizer unchanged, making it composable with existing optimizers. Importantly, optimizer-side transformations act only after the per-example gradients have been aggregated, whereas LISC can alter the direction selected by the aggregation itself. Appendix C makes this distinction precise for Muon through a simple rank-one construction, where Muon alone cannot rotate the aggregated feature direction, while LISC reweighting can rotate it before Muon is applied and yields a strictly larger first-order improvement in test loss.

Training procedure. Figure 2 summarizes the overall LISC pipeline. Given the mini-batch, LISC constructs the sample-gradient Gram matrix, computes the spectral weights, and uses the resulting weighted loss with the downstream optimizer. The complete algorithm is provided in Algorithm 1.

5 Experiments

We evaluate how LISC affects generalization throughout training, how restricting the measurement geometry trades performance for computational cost, and whether its gains arise from smaller updates

Table 1: **Benchmark performance.** Held-out accuracy or exact match (%; higher is better) over three seeds. Checkpoint triplets denote epochs, except for MMLU (optimizer updates). Base uses the listed optimizer.

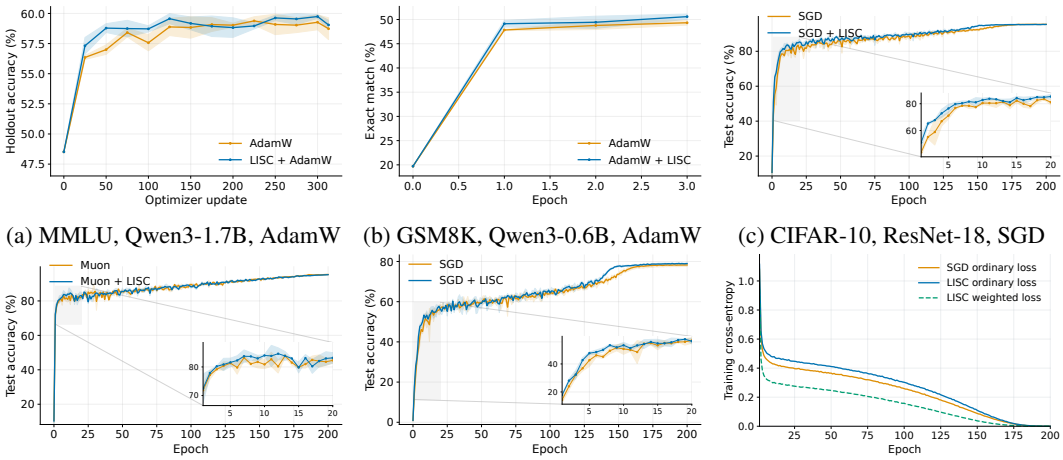
Setting	Method	Early	Middle	AUC↑	Final
Vision					
CIFAR-10	Base	71.14 ± 2.48	81.09 ± 1.88	71.74 ± 0.88	**95.40** ± 0.09
ResNet-18 (SGD)	Aligned-MTL	73.44 ± 2.07	84.59 ± 0.76	73.89 ± 1.06	90.93 ± 0.22
(5, 20, 200)	LISC	**76.48** ± 2.80	**85.36** ± 1.63	**75.48** ± 1.04	95.37 ± 0.12
CIFAR-10	Base	81.33 ± 0.31	82.30 ± 1.00	78.93 ± 0.91	**95.35** ± 0.13
ResNet-18 (Muon)	Aligned-MTL	81.30 ± 0.98	82.78 ± 1.19	79.20 ± 0.83	93.28 ± 1.27
(5, 20, 200)	LISC	**81.59** ± 1.46	**83.17** ± 1.82	**80.01** ± 0.88	95.26 ± 0.16
CIFAR-10	Base	59.51 ± 1.08	**79.82** ± 0.55	69.38 ± 0.94	**93.52** ± 0.07
CCT-7 (AdamW)	Aligned-MTL	58.29 ± 0.93	76.73 ± 1.04	67.85 ± 0.90	91.60 ± 0.95
(5, 20, 300)	LISC	**62.07** ± 0.75	79.44 ± 0.71	**70.36** ± 1.02	93.32 ± 0.15
CIFAR-100	Base	42.68 ± 2.47	**56.72** ± 1.99	45.22 ± 0.82	78.26 ± 0.73
ResNet-18 (SGD)	Aligned-MTL	35.80 ± 0.51	49.54 ± 0.76	39.79 ± 0.58	74.09 ± 0.35
(5, 20, 200)	LISC	**47.70** ± 0.86	56.24 ± 1.98	**47.41** ± 0.78	**78.97** ± 0.26
CIFAR-100	Base	52.35 ± 2.89	56.70 ± 0.67	51.45 ± 0.74	78.15 ± 0.12
ResNet-18 (Muon)	Aligned-MTL	43.80 ± 1.21	49.85 ± 1.03	45.59 ± 0.52	75.84 ± 1.06
(5, 20, 200)	LISC	**52.53** ± 1.56	**58.05** ± 0.86	**52.25** ± 0.81	**78.20** ± 0.10
ImageNet-1K	Base	41.65 ± 1.36	51.55 ± 1.72	42.03 ± 0.76	76.07 ± 0.17
ResNet-50 (SGD)	Aligned-MTL	36.81 ± 0.40	42.96 ± 0.68	38.63 ± 0.48	74.80 ± 1.36
(5, 20, 90)	LISC	**43.63** ± 1.77	**51.90** ± 1.89	**43.29** ± 0.82	**76.21** ± 0.19
ImageNet-1K	Base	57.37 ± 0.61	**65.56** ± 1.10	56.35 ± 0.93	74.33 ± 0.81
ResNet-50 (Muon)	Aligned-MTL	53.79 ± 0.68	62.10 ± 0.74	52.83 ± 0.96	74.01 ± 1.21
(5, 20, 90)	LISC	**57.93** ± 1.36	65.30 ± 1.62	**57.06** ± 0.89	**74.85** ± 0.26
Language fine-tuning					
SST-2	Base	95.03 ± 0.67	95.53 ± 0.11	82.47 ± 1.12	95.76 ± 0.23
DeBERTa-v3-base (AdamW)	Aligned-MTL	93.73 ± 0.81	93.46 ± 2.20	81.05 ± 1.15	93.43 ± 0.98
(1, 2, 3)	LISC	**95.22** ± 0.35	**95.60** ± 0.18	**82.57** ± 1.20	**95.83** ± 0.26
GSM8K	Base	47.84 ± 0.13	48.77 ± 0.84	43.90 ± 0.57	49.33 ± 0.49
Qwen3-0.6B (AdamW)	Aligned-MTL	48.88 ± 0.23	49.36 ± 0.96	44.40 ± 0.58	49.89 ± 1.58
(1, 2, 3)	LISC	**49.13** ± 0.68	**49.43** ± 1.25	**44.60** ± 0.61	**50.59** ± 0.54
CommonsenseQA	Base	69.34 ± 1.49	71.72 ± 0.80	71.10 ± 0.65	71.64 ± 1.15
Qwen3-0.6B (AdamW)	Aligned-MTL	65.22 ± 0.67	66.50 ± 0.85	68.06 ± 0.74	68.66 ± 1.38
(1, 2, 3)	LISC	**70.11** ± 0.81	**71.99** ± 0.46	**71.46** ± 0.78	**71.73** ± 0.55
MMLU	Base	56.35 ± 0.16	57.57 ± 0.67	57.79 ± 0.63	58.73 ± 0.91
Qwen3-1.7B (AdamW)	Aligned-MTL	56.92 ± 0.32	57.38 ± 0.35	57.83 ± 0.78	58.81 ± 0.48
(25, 100, 313)	LISC	**57.32** ± 0.69	**58.72** ± 0.26	**58.25** ± 0.70	**59.05** ± 0.39

or changes in update direction. Comparisons with sample-wise Aligned-MTL and sensitivity analyses further examine the role of spectral reweighting.

5.1 Experimental setup

Vision experiments cover ResNet-18/50 (He et al., 2016) and CCT-7 (Hassani et al., 2021) on CIFAR-10/100 (Krizhevsky, 2009) and ImageNet-1K (Deng et al., 2009). Language experiments include DeBERTa-v3-base (He et al., 2023) on SST-2 (Socher et al., 2013), Qwen3 (Yang et al., 2025) with rank-16 LoRA (Hu et al., 2022) on GSM8K (Cobbe et al., 2021), CommonsenseQA (Talmor et al., 2019), and MMLU (Hendrycks et al., 2021), and GPT-2 Small (Radford et al., 2019) pretraining on FineWeb (Penedo et al., 2024). We compare against the underlying optimizer (Base: SGD, AdamW, or Muon) and sample-wise Aligned-MTL (Senushkin et al., 2023), using the same measurement coordinates as LISC.

Unless ablated, we use $\gamma = 0.03$, motivated by the sensitivity analysis in Table 6, and recompute weights for each mini-batch. Measurement coordinates are the classifier head for vision, the trainable



(a) MMLU, Qwen3-1.7B, AdamW (b) GSM8K, Qwen3-0.6B, AdamW (c) CIFAR-10, ResNet-18, SGD

(d) CIFAR-10, ResNet-18, Muon (e) CIFAR-100, ResNet-18, SGD (f) Train loss, CIFAR-10, SGD

Figure 3: **LISC improves early generalization without necessarily reducing training loss faster.** (a–e) Test performance for representative language and vision settings; (f) training loss on CIFAR-10 with ResNet-18 and SGD.

adapters for Qwen3, and the final MLP output projection for GPT-2. Base and LISC share batch sizes, learning rates, and schedules; selected weight decay differs in three settings. Appendices B.2 and B.3 provide implementation details and configurations, while Appendix B.7 compares matched hyperparameter configurations.

5.2 Results

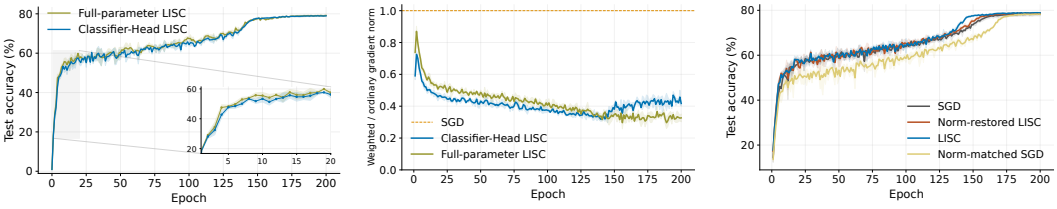
Results are means $\pm$ sample standard deviations over three seeds. We report fixed checkpoints, normalized area under the held-out performance curve (AUC), and the final scheduled checkpoint. AUC covers the first 10% of training for vision and the full trajectory for language fine-tuning tasks, which have shorter training horizons (Appendix B.1). Improvement refers to reported means, and acceleration to generalization per optimization step. Appendix B.9 reports wall-clock runtime and peak-memory overhead.

Consistent early generalization. As shown in Table 1, LISC achieves the highest mean early performance and AUC in all 11 settings. Relative to Base, the largest early gains are on CIFAR-10 and CIFAR-100 with SGD: $+5.34$ and $+5.02$ percentage points (pp), respectively. On CIFAR-100, AUC increases from 45.22 ± 0.82 to 47.41 ± 0.78. Intermediate performance improves in 8 of 11 settings. Figure 3 shows representative trajectories.

Preserved or improved final generalization. LISC improves mean final performance over Base in 8 of 11 settings. The three exceptions are on CIFAR-10, with differences of -0.03, -0.09, and -0.20 pp. Language fine-tuning results are especially consistent. All four language fine-tuning tasks improve at every reported checkpoint, including a 1.26 pp final gain on GSM8K. LISC also exceeds Aligned-MTL's mean final performance in all 11 settings. GPT-2 pretraining shows small reductions in mean held-out loss at all reported checkpoints (Appendix B.6).

Comparison with spectral equalization. LISC exceeds sample-wise Aligned-MTL in mean early performance, AUC, and final performance across all 11 settings. These results strongly favor LISC's damped spectral reweighting over the evaluated equalization baseline, supporting the value of retaining differences in spectral strength for both early learning and final generalization.

Optimization objective. LISC explicitly optimizes the instantaneous objective $\sum_i w_i \ell_i$, whose detached weights change with the mini-batch and model state. Although $\sum_i w_i = 1$, different example weights produce a different objective value. The CIFAR-10 loss comparison illustrates that improved held-out performance can coexist with slower reduction of the uniformly weighted training



(a) Full- vs. head-parameter LISC. (b) Gradient-norm contraction. (c) Norm-matched test accuracy.

Figure 4: **LISC contracts the gradient norm.** The LISC transformation reduces the gradient magnitude. Increasing the step size to compensate for this reduction preserves performance. Conversely, reducing the SGD step size so that its gradient-step magnitude matches that induced by LISC leads to a drop in performance.

objective. Thus, generalization gains need not coincide with faster minimization of the reference empirical loss.

5.3 Ablation Study

Restricting the measurement geometry. Figure 4a compares full-parameter and classifier-head measurement on CIFAR-100. Full-parameter LISC yields stronger gains early in training, while the substantially cheaper classifier-head variant catches up later. This difference comes at a large computational cost: full-parameter LISC requires 1828.44 ± 4.46 minutes, compared with 28.44 ± 0.46 minutes for classifier-head LISC in the CIFAR-100 SGD setting. Given this cost–performance trade-off, we use classifier-head geometry by default in the vision experiments.

Update direction versus magnitude. Figure 4b shows that LISC contracts the aggregated gradient in the evaluated setting. Although Proposition 4 guarantees contraction for the raw transformation, contraction after clipping and normalization is an empirical observation. We therefore test whether the performance gains can be explained simply by the smaller effective update magnitude. We consider two norm-matching controls: (i) increasing the LISC step size so that its update norm matches the SGD baseline, and (ii) decreasing the SGD step size so that its update norm matches that of LISC. Figure 4c shows that compensating for LISC's contraction preserves both its early-learning advantage and final performance. In contrast, reducing the SGD step size does not reproduce the early acceleration and yields worse final performance. These controls indicate that gradient contraction alone is insufficient to explain the gains, and instead support the role of the update direction induced by LISC's sample reweighting.

6 Conclusion

Sample aggregation shapes optimization by determining how per-example learning signals combine into the gradient received by the optimizer. We introduced LISC to control this choice through the evolving geometry of mini-batch gradients. Its damped spectral filter reduces the relative influence of strongly represented modes, and its raw transformation establishes an exact connection between sample reweighting and local inverse-square-root second-moment preconditioning. Empirically, across all 11 considered benchmark settings, LISC accelerates early performance with competitive or improved final generalization. Norm-matching controls indicate that smaller updates alone do not explain these gains, while comparisons with sample-wise Aligned-MTL support the benefit of finite damping over complete spectral equalization.

More broadly, our results suggest that *sample aggregation itself is an optimization operator*, with LISC providing one concrete realization. Identifying other optimization geometries that can be realized through sample weights could deepen the connection between data weighting and optimizer design. Faster approximations to the matrix-function action (Amsel et al., 2024; Anil et al., 2021) could make these methods more efficient and scalable. By connecting adaptive data curricula with preconditioning, LISC offers a new perspective for jointly designing sample weighting and optimizer transformations.

AI USE STATEMENT

We used generative AI tools to assist with refining hypotheses and mathematical formulations, designing experiments, writing code, analyzing and interpreting experimental results, discovering relevant literature, preparing scientific figures, and drafting and editing portions of the manuscript. In particular, we used large language models to help identify and develop the illustrative construction presented in Appendix C. We also used these tools to help identify and address issues in mathematical claims and their supporting arguments. We independently reviewed and verified all AI-assisted material, checked the mathematical arguments included in the paper, and verified the reported experimental results against the underlying logs and artifacts. We take full responsibility for the final text, claims, code, figures, mathematical results, and experimental findings presented in this work.

ETHICS STATEMENT

Our work proposes a general-purpose training method and evaluates it on publicly available benchmark datasets and models. We do not collect new data, conduct human-subject studies, or introduce datasets containing additional personal or sensitive information. As with other methods that improve neural-network training, LISC could be applied to downstream systems with application-specific societal impacts; such impacts depend primarily on the models, data, and deployment settings in which it is used. We are not aware of additional ethical concerns specific to the proposed method.

REPRODUCIBILITY STATEMENT

We provide the full specification of LISC, including its weighting rule, measurement geometry, and practical implementation, in the main paper, with pseudocode and computational details in the appendix. The appendix additionally contains complete proofs, hyperparameter grids, ablations, seed-level comparisons, and implementation details. We provide the source code used for our experiments as an anonymized supplementary archive. For baseline–LISC comparisons, we report the corresponding training configurations and evaluation protocols, and use matched architectures, data, augmentation, and learning-rate schedules where appropriate. Together, these materials are intended to enable independent reproduction of both the theoretical and empirical results.

REFERENCES

Amirhesam Abedsoltan, Siyuan Ma, Parthe Pandit, and Misha Belkin. Fast training of large kernel models with delayed projections. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.neurips.cc/paper_files/paper/2025/hash/42aae0e655c77d93edad9171ad9f4717-Abstract-Conference.html.

Noah Amsel, Tyler Chen, Anne Greenbaum, Cameron Musco, and Christopher Musco. Nearly optimal approximation of matrix functions by the lanczos method. In *Advances in Neural Information Processing Systems*, volume 37, pp. 139823–139853. Curran Associates, Inc., 2024.

Rohan Anil, Vineet Gupta, Tomer Koren, Kevin Regan, and Yoram Singer. Scalable second order optimization for deep learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/arpit17a.html.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48. ACM, 2009. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Powei Chang, Jinpeng Zhang, Chaoqun Sun, MiniWell Tsao, Lianrui Li, Jianxiang Xiang, Chenyu Wang, Yukang Gao, and Dongying Kong. SALT: When more rollouts don't help in group-based

policy optimization and how to make them matter. *arXiv preprint arXiv:2606.05800*, 2026. URL https://arxiv.org/abs/2606.05800.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Advait Gadhikar, Tom Jacobs, Chao Zhou, and Rebekka Burkholz. Sign-in to the lottery: Reparameterizing sparse training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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

Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *arXiv preprint arXiv:2104.05704*, 2021. URL https://arxiv.org/abs/2104.05704.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. URL https://arxiv.org/abs/2203.15556.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Tom Jacobs, Advait Gadhikar, Celia Rubio-Madrigal, and Rebekka Burkholz. Hyperbolic aware minimization: Implicit bias for sparsity. In *The Fourteenth International Conference on Learning Representations*, 2026a.

Tom Jacobs, Rohan Jain, and Rebekka Burkholz. Horst: Composing optimizer geometries for sparse transformer training. In *Workshop on High-dimensional Learning Dynamics (HiLD) @ ICML*, 2026b. URL https://arxiv.org/abs/2605.21104.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361.

Krishnateja Killamsetty, Durga S, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5464–5474. PMLR, 2021. URL https://proceedings.mlr.press/v139/killamsetty21a.html.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 5637–5664, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23, pp. 1189–1197, 2010. URL https://proceedings.neurips.cc/paper/2010/hash/e57c6b956a6521b28495f2886ca0977a-Abstract.html.

Frederik Kunstner, Lukas Balles, and Philipp Hennig. Limitations of the empirical fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/46a558d97954d0692411c861cf78ef79-Abstract.html.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025. URL https://arxiv.org/abs/2502.16982.

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

Chao Ma, Wenbo Gong, Meyer Scetbon, and Edward Meeds. SWAN: SGD with normalization and whitening enables stateless LLM training. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 41907–41942, 2025. URL https://proceedings.mlr.press/v267/ma25g.html.

Siyuan Ma and Mikhail Belkin. Diving into the shallows: A computational perspective on large-scale shallow learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.neurips.cc/paper_files/paper/2017/hash/bf424cb7b0dea050a42b9739eb261a3a-Abstract.html.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020. URL https://www.jmlr.org/papers/v21/17-678.html.

Nimrah Mustafa and Rebekka Burkholz. Dynamic rescaling for training GNNs. In *Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://papers.neurips.cc/paper/2021/hash/ac56f8fe9eea3e4a365f29f0f1957c55-Abstract.html.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849, 2024. doi: 10.52202/079017-0970. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/0987b8b338d6c90bbedd8631bc499221-Abstract.html.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, and Yang You. InfoBatch: Lossless training speed up by unbiased dynamic data pruning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/ad1d7a4df30a9c0c46b387815a774a84-Abstract-Conference.html.

Pierre Quinton and Valérian Rey. Jacobian descent for multi-objective optimization. *arXiv preprint arXiv:2406.16232*, 2024. URL https://arxiv.org/abs/2406.16232. Version 3 updated February 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 2019. URL https://proceedings.mlr.press/v97/rahaman19a.html.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4334–4343. PMLR, 2018. URL https://proceedings.mlr.press/v80/ren18a.html.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 8346–8356, 2020.

Karthik Abinav Sankararaman, Soham De, Zheng Xu, W. Ronny Huang, and Tom Goldstein. The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8469–8479. PMLR, 2020. URL https://proceedings.mlr.press/v119/sankararaman20a.html.

Sunny Sanyal, Hayden Prairie, Rudrajit Das, Ali Kavis, and Sujay Sanghavi. Upweighting easy samples in fine-tuning mitigates forgetting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 52922–52957, 2025. URL https://proceedings.mlr.press/v267/sanyal25a.html.

Dmitry Senushkin, Nikolay Patakin, Arseny Kuznetsov, and Anton Konushin. Independent component alignment for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20083–20093, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Senushkin_Independent_Component_Alignment_for_Multi-Task_Learning_CVPR_2023_paper.html.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6cfe0e6127fa25df2a0ef2ae1067d915-Abstract.html.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-Weight-Net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/e58cc5ca94270acaceed13bc82dfedf7-Abstract.html.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642. Association for Computational Linguistics, 2013. URL https://aclanthology.org/D13-1170/.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9275–9293, 2020. doi: 10.18653/v1/2020.emnlp-main.746.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. SOAP: Improving and stabilizing Shampoo using Adam for language modeling. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IDxZhXrpNf.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/dcba6be91359358c2355cd920da3fcbd-Abstract-Conference.html.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Dong Yin, Ashwin Pananjady, Max Lam, Dimitris Papailiopoulos, Kannan Ramchandran, and Peter Bartlett. Gradient diversity: a key ingredient for scalable distributed learning. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1998–2007. PMLR, 2018. URL https://proceedings.mlr.press/v84/yin18a.html.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

A Appendix

A.1 LISC Algorithm and Implementation Details

Algorithm 1 summarizes one LISC training step. LISC first measures interactions between per-example gradients, attenuates strongly represented spectral modes, and converts the resulting coefficients into nonnegative sample weights. These weights are then used with the original training loss and optimizer.

Algorithm 1 Local Inverse Spectral Curriculum (LISC)

Require: Mini-batch $\mathcal{B}$, parameters θ, measurement coordinates ϕ, reference weights $m \in \Delta_b$, damping $\gamma > 0$, optimizer $\mathcal{O}$.

1: Compute per-example losses $\ell_1, \dots, \ell_b$ from the current batch
2: Compute measurement gradients $a_i = \nabla_\phi \ell_i$ and stack $A = [a_1^\top; \dots; a_b^\top]$
3: Form the sample-gradient Gram matrix $K = AA^\top$
4: Compute $K = U \operatorname{diag}(\lambda_1, \dots, \lambda_b) U^\top$ with $\lambda_1 = \lambda_{\max}(K)$
5: Set $\tau = \gamma \lambda_1$
6: Compute spectral factors $h_j = \sqrt{\frac{\tau}{\lambda_j + \tau}}$
7: Compute raw coefficients $\widehat{w} = U \operatorname{diag}(h_1, \dots, h_b) U^\top m$
8: Convert to sample weights $w = [\widehat{w}]_+ / \left(\mathbf{1}^\top [\widehat{w}]_+\right)$
9: Treat w as constant during differentiation
10: Form the weighted loss $L_{\mathrm{LISC}} = \sum_{i=1}^b w_i \ell_i$
11: Update θ using optimizer $\mathcal{O}$ on $\nabla_\theta L_{\mathrm{LISC}}$

Equivalently, after eigendecomposing K,

$$\widehat{w} = U \operatorname{diag}\left(\sqrt{\frac{\tau}{\lambda_1 + \tau}}, \dots, \sqrt{\frac{\tau}{\lambda_b + \tau}} \right) U^\top m.$$

In implementation, eigenvalues that are slightly negative due to floating-point roundoff are clamped to zero. The spectral stage requires $O(b^2)$ storage and $O(b^3)$ time for a dense eigendecomposition. The cost of constructing K depends on the measurement coordinates. For classifier-head geometry, equation 3 avoids materializing full per-example parameter gradients; for adapter geometry, the cost is determined by the trainable adapter dimension and the implementation of per-example gradients.

A.2 Proof of Lemma 1

For weights held fixed during differentiation, the reference mini-batch objective is

$$F_t(\phi) = m^\top \boldsymbol{\ell}(\phi),$$

and

$$\nabla_\phi \left(w^\top \boldsymbol{\ell}(\phi) \right) = A^\top w = g_w.$$

Using the first-order expansion

$$\boldsymbol{\ell}(\phi + \delta\phi) = \boldsymbol{\ell}(\phi) + A\delta\phi + o(\|\delta\phi\|_2),$$

with $\delta\phi = -\eta g_w$, and noting that g_w is fixed at the current iterate, gives

$$\boldsymbol{\ell}(\phi - \eta g_w) = \boldsymbol{\ell}(\phi) - \eta A g_w + o(\eta) = \boldsymbol{\ell}(\phi) - \eta K w + o(\eta). \tag{4}$$

Multiplying equation 4 on the left by $m^\top$ yields

$$F_t(\phi - \eta g_w) - F_t(\phi) = -\eta m^\top K w + o(\eta)$$

Finally, since $g = A^\top m$ and $g_w = A^\top w$,

$$m^\top K w = m^\top A A^\top w = \langle A^\top m, A^\top w \rangle = \langle g, g_w \rangle,$$

which gives

$$F_t(\phi - \eta g_w) - F_t(\phi) = -\eta \langle g, g_w \rangle + o(\eta)$$

A.3 Proof of Proposition 2

Recall that $K = AA^\top$ is the sample-gradient Gram matrix and $S = A^\top A$ is the corresponding parameter-space gradient second-moment operator. Let $A = U\Sigma V^\top$ be the compact singular value decomposition of A, with rank r. Then

$$K = AA^\top = U\Sigma^2 U^\top, \qquad S = A^\top A = V\Sigma^2 V^\top,$$

so K and S share the same nonzero eigenvalues $\lambda_j = \sigma_j^2$, $j = 1, \dots, r$.

Complete U to an orthonormal basis $[U\ U_\perp]$ of $\mathbb{R}^b$. Then

$$AA^\top = [U\ U_\perp] \begin{bmatrix} \Sigma^2 & 0 \\ 0 & 0 \end{bmatrix} [U\ U_\perp]^\top$$

Applying the spectral function f_τ therefore gives

$$f_\tau(AA^\top) = U f_\tau(\Sigma^2) U^\top + f_\tau(0) U_\perp U_\perp^\top$$

Since $f_\tau(0) = 1$ and $U_\perp U_\perp^\top = I - UU^\top$, we obtain

$$f_\tau(AA^\top) = U f_\tau(\Sigma^2) U^\top + (I - UU^\top)$$

Moreover, $I - UU^\top$ projects onto $\ker(A^\top)$, so $A^\top(I - UU^\top) = 0$. Hence

$$A^\top f_\tau(AA^\top) = V\Sigma f_\tau(\Sigma^2) U^\top = V f_\tau(\Sigma^2)\Sigma U^\top,$$

where the second equality follows because Σ and $f_\tau(\Sigma^2)$ are diagonal and therefore commute.

Similarly, completing V to an orthonormal basis gives

$$f_\tau(A^\top A) = V f_\tau(\Sigma^2) V^\top + (I - VV^\top),$$

and $(I - VV^\top)A^\top = 0$. Therefore,

$$A^\top f_\tau(AA^\top) = f_\tau(A^\top A) A^\top. \tag{5}$$

For any positive semidefinite matrix with full eigendecomposition $S = W\Lambda W^\top$, the corresponding spectral matrix function is obtained by applying f_τ to its eigenvalues:

$$f_\tau(S) = W f_\tau(\Lambda) W^\top = W \left(I + \frac{\Lambda}{\tau}\right)^{-1/2} W^\top = \left(I + \frac{S}{\tau}\right)^{-1/2}. \tag{6}$$

The last equality follows from $I + \frac{S}{\tau} = W\left(I + \frac{\Lambda}{\tau}\right) W^\top$.

Combining Equations (2), (5) and (6)

$$\widehat{g} = A^\top f_\tau(K) m = f_\tau(S) A^\top m = (I + S/\tau)^{-1/2} g.$$

Using the decomposition of g from equation 1,

$$\widehat{g} = \sum_{j=1}^{r} \sqrt{\lambda_j}\, f_\tau(\lambda_j) c_j v_j.$$

For the optimization interpretation, define

$$M = (I + S/\tau)^{1/2}.$$

Since $S \succeq 0$ and $\tau > 0$, we have $M \succ 0$. Hence

$$Q(\delta) = g^\top \delta + \frac{1}{2\eta} \delta^\top M \delta$$

is strictly convex. Its unique stationary point satisfies

$$g + \frac{1}{\eta} M\delta = 0,$$

and therefore

$$\delta_* = -\eta M^{-1} g = -\eta (I + S/\tau)^{-1/2} g = -\eta \widehat{g}.$$

Strict convexity implies that δ_* is the unique global minimizer.

A.4 Proof of Theorem 3

Let $a_j = f_\tau(\lambda_j)^2$. Because $\lambda_1 \geq \cdots \geq \lambda_r$ and f_τ is decreasing, $a_1 \leq a_2 \leq \cdots \leq a_r$.

If $r = \operatorname{rank}(A)$, for a fixed $k < r$, define

$$X = \sum_{i \leq k} E_i, \qquad Y = \sum_{j > k} E_j,$$
$$X' = \sum_{i \leq k} a_i E_i, \qquad Y' = \sum_{j > k} a_j E_j.$$

Since $g \neq 0$ and $a_j > 0$, the relevant denominators are positive.

Cross multiplication gives

$$\widehat{C}_k \leq C_k \quad \Longleftrightarrow \quad X'Y \leq XY'.$$

The difference between the two sides is

$$XY' - X'Y = \sum_{i \leq k} \sum_{j > k} E_i E_j (a_j - a_i) \geq 0.$$

Therefore $\widehat{C}_k \leq C_k$. The inequality is strict if there is at least one pair $i \leq k < j$ with $E_i E_j > 0$ and $\lambda_i > \lambda_j$, because in that case the strict monotonicity of f_τ gives $a_j > a_i$. For $k = r$, both cumulative shares equal one.

Therefore,

$$\widehat{E}_j = E_j \frac{\tau}{\lambda_j + \tau} = c_j^2 \frac{\tau \lambda_j}{\lambda_j + \tau} \leq \tau c_j^2.$$

Remark: why energy entropy need not increase. Theorem 3 concerns cumulative energy ordered by decreasing Gram eigenvalue; it is not a majorization result after re-sorting modes by energy. For example, consider

$$\lambda_1 = 4, \qquad \lambda_2 = 1, \qquad \tau = 0.03,$$

with original normalized mode energies $(0.2, 0.8)$. The energy multipliers are

$$\left(\frac{0.03}{4.03}, \frac{0.03}{1.03} \right) \approx (0.00744, 0.02913),$$

giving filtered proportions

$$\left(\frac{103}{1715}, \frac{1612}{1715} \right) \approx (0.060, 0.940).$$

The fraction of energy in the leading-eigenvalue mode decreases, as guaranteed by the theorem, even though the entropy of the two-mode energy distribution also decreases. Thus, Theorem 3 does not assert that the entropy or effective rank of the energy distribution must increase for every batch. These statistics depend on the original overlaps c_j and can behave differently when the unfiltered gradient is already concentrated in lower-eigenvalue modes.

A.5 Energy spreading and effective rank

Define

$$p_j = \frac{E_j}{\sum_i E_i}, \qquad \widehat{p}_j = \frac{E_j \tau/(\lambda_j + \tau)}{\sum_i E_i \tau/(\lambda_i + \tau)}, \qquad r_E(p) = \exp\left(- \sum_j p_j \log p_j \right).$$

Proposition 5 (Energy effective rank under majorization). *Let $p^{\downarrow}$ and $\widehat{p}^{\downarrow}$ denote the energy fractions sorted in decreasing order, independently of eigenvalue order. If*

$$\sum_{j=1}^{k} p_j^{\downarrow} \geq \sum_{j=1}^{k} \widehat{p}_j^{\downarrow} \quad \text{for every } k < r, \tag{7}$$

then $r_E(\widehat{p}) \geq r_E(p)$, with strict inequality unless the two sorted energy distributions coincide. A sufficient condition for equation 7 is that both E_j and $E_j\tau/(\lambda_j+\tau)$ are nonincreasing in the original, decreasing-eigenvalue order.

Proof. The strictly positive spectral multipliers preserve the support of the energy distribution. Discard any shared zero entries and write $x = p^{\downarrow}$, $y = \widehat{p}^{\downarrow}$ on the remaining n entries. Set $D_k = \sum_{j\leq k}(x_j - y_j)$. By assumption, $D_k \geq 0$ for $k < n$, and $D_n = 0$. Summation by parts gives

$$\sum_{j=1}^{n}(x_j - y_j)\log y_j = \sum_{k=1}^{n-1} D_k(\log y_k - \log y_{k+1}) \geq 0.$$

For $H(p) = -\sum_j p_j \log p_j$, concavity yields

$$\begin{aligned} H(x) &\leq H(y) + \nabla H(y)^{\top}(x - y) \\ &= H(y) - \sum_j (x_j - y_j)\log y_j \leq H(y). \end{aligned}$$

The first inequality is strict if $x \neq y$, since entropy is strictly concave on the positive simplex. Sorting leaves entropy unchanged, and exponentiation establishes the rank claim.

For the sufficient condition, suppose the two energy sequences are nonincreasing in eigenvalue order. Their normalized versions are then already sorted. Put $h_j = \tau/(\lambda_j + \tau)$, which is nondecreasing in j. For any cutoff with positive energy on both sides, its p-weighted mean on the prefix is no larger than its weighted mean on the suffix. If the two means are $a \leq b$ and $P_k = \sum_{j\leq k} p_j$, then

$$\sum_{j\leq k} \widehat{p}_j = \frac{P_k a}{P_k a + (1 - P_k)b} \leq P_k.$$

The zero-energy cases are immediate. This proves majorization and completes the proof. □

A.6 Proof of Proposition 4

Let $a_j = f_\tau(\lambda_j)$. When $\tau = \gamma\lambda_1$, the eigenvalue ordering $0 < \lambda_j \leq \lambda_1$ implies that

$$\sqrt{\frac{\gamma}{1+\gamma}} \leq a_j \leq 1.$$

Using the spectral decomposition $E_j = \lambda_j c_j^2$, we can express the inner product and the squared norm of the update as:

$$\langle g, \widehat{g}\rangle = \sum_{j=1}^{r} E_j a_j, \qquad \|\widehat{g}\|_2^2 = \sum_{j=1}^{r} E_j a_j^2.$$

Since $a_j^2 \leq a_j \leq 1$, we immediately obtain the bounds

$$\|\widehat{g}\|_2^2 \leq \langle g, \widehat{g}\rangle \leq \|g\|_2^2,$$

which guarantees that $\|\widehat{g}\|_2 \leq \|g\|_2$. Likewise, because $a_j \geq \sqrt{\frac{\gamma}{1+\gamma}} > 0$ and $g \neq 0$, we have

$$\langle g, \widehat{g}\rangle \geq \sqrt{\frac{\gamma}{1+\gamma}}\|g\|_2^2 > 0,$$

which confirms $\widehat{g} \neq 0$ and establishes strict positive alignment.

Finally, suppose the reference objective restricted to ϕ is L-smooth. The standard descent lemma gives

$$F_t(\phi - \eta\widehat{g}) \le F_t(\phi) - \eta\langle g, \widehat{g}\rangle + \frac{L\eta^2}{2}\|\widehat{g}\|_2^2.$$

Substituting the upper bound $\|\widehat{g}\|_2^2 \le \langle g, \widehat{g}\rangle$ yields

$$F_t(\phi - \eta\widehat{g}) \le F_t(\phi) - \eta\left(1 - \frac{L\eta}{2}\right)\langle g, \widehat{g}\rangle.$$

For a step size $0 < \eta < 2/L$, the term $(1 - \frac{L\eta}{2})$ is positive. Substituting our lower bound for the inner product, $\langle g, \widehat{g}\rangle \ge \sqrt{\frac{\gamma}{1+\gamma}}\|g\|_2^2$, directly yields the descent bound in the proposition.

Remark. The smoothness condition above is local and conditional. It is not implied by the first-order expansion in Lemma 1, nor do we require global smoothness of the neural network objective.

A.7 Conditioning of the filtered operator

Proposition 2 characterizes the transformation of a particular reference gradient g. We next characterize the spectrum of the corresponding raw filtered operator independently of the reference weights m. Define

$$\widetilde{A} = f_\tau(K)A, \qquad \widetilde{K} = \widetilde{A}\widetilde{A}^\top,$$

so that $\widehat{g} = \widetilde{A}^\top m$. For a nonzero positive-semidefinite matrix M, let $\kappa_+(M) = \lambda_{\max}(M)/\lambda^+_{\min}(M)$ denote its condition number restricted to its positive spectrum.

Proposition 6 (Spectral compression of the raw filtered Gram operator)**.** *The positive eigenvalues of $\widetilde{K}$ are*

$$\widetilde{\lambda}_j = \frac{\tau\lambda_j}{\lambda_j + \tau}.$$

Consequently, its algebraic rank is unchanged, and

$$\kappa_+(\widetilde{K}) = \kappa_+(K)\frac{\lambda_r + \tau}{\lambda_1 + \tau} \le \kappa_+(K).$$

The inequality is strict when $\lambda_1 > \lambda_r$. For $\tau = \gamma\lambda_1$, this becomes

$$\kappa_+(\widetilde{K}) = \frac{1 + \gamma\kappa_+(K)}{1 + \gamma}.$$

Furthermore, the entropy effective rank of the positive eigenvalues of $\widetilde{K}$ is at least that of K, with strict inequality if the positive eigenvalues of K are not all equal.

Proof. Matrix functions of K commute with K, so

$$\widetilde{K} = f_\tau(K)Kf_\tau(K) = \tau K(K + \tau I)^{-1}.$$

The scalar function $h(\lambda) = \tau\lambda/(\lambda + \tau)$ is strictly increasing on $[0, \infty)$ and positive for $\lambda > 0$. Thus it preserves the positive-spectrum ordering and algebraic rank. Taking the ratio of its extreme positive eigenvalues gives

$$\begin{aligned}\kappa_+(\widetilde{K}) &= \frac{\tau\lambda_1/(\lambda_1 + \tau)}{\tau\lambda_r/(\lambda_r + \tau)} \\ &= \kappa_+(K)\frac{\lambda_r + \tau}{\lambda_1 + \tau}.\end{aligned}$$

The inequality and its strictness follow immediately. Substituting $\tau = \gamma\lambda_1$ gives $(1 + \gamma\kappa_+(K))/(1 + \gamma)$.

For spectral entropy, normalize the positive eigenvalues to form $v_j = \lambda_j/\sum_i \lambda_i$ and $\widetilde{v}_j = h(\lambda_j)/\sum_i h(\lambda_i)$. Both are sorted in decreasing eigenvalue order. The argument in Proposition 5, with $E_j = \lambda_j$, shows that v majorizes $\widetilde{v}$ and hence that spectral entropy does not decrease. The distributions coincide when $h(\lambda_j)/\lambda_j = \tau/(\lambda_j + \tau)$ is constant on the positive spectrum, that is, when all positive eigenvalues are equal. This also proves strictness. □

A.8 CONVERGENCE UNDER CURVATURE-MATCHED QUADRATIC SPECIALIZATION

The previous results concern the gradient second-moment geometry used by LISC and do not require it to equal curvature. To isolate the optimization consequence when these operators do coincide, we consider the idealized case of a fixed quadratic objective with $S = H$.

Theorem 7 (Convergence in a curvature-matched quadratic model)**.** *Consider $Q(\phi) = \frac{1}{2}(\phi - \phi_*)^\top H(\phi - \phi_*)$, where $H \succ 0$. Assume the measurement second moment is frozen and equals H, so that the raw update uses*

$$P = (I + H/\tau)^{-1/2}, \qquad \phi_{t+1} = \phi_t - \eta P \nabla Q(\phi_t), \qquad \tau > 0.$$

Let $\mu_{\max}$ and $\mu_{\min}$ be the extreme eigenvalues of H, and set

$$\nu_j = \mu_j \sqrt{\frac{\tau}{\mu_j + \tau}}, \qquad \kappa_{\mathrm{opt}} = \kappa(H) \sqrt{\frac{\mu_{\min} + \tau}{\mu_{\max} + \tau}}.$$

At $\eta_* = 2/(\nu_{\max} + \nu_{\min})$, the iterates satisfy

$$Q(\phi_t) \le \left(\frac{\kappa_{\mathrm{opt}} - 1}{\kappa_{\mathrm{opt}} + 1} \right)^{2t} Q(\phi_0).$$

If $\kappa(H) > 1$, this worst-case contraction factor is strictly smaller than that of ordinary gradient descent using its own optimal constant step size. As $\tau \downarrow 0$, κ_{opt} approaches $\sqrt{\kappa(H)}$.

Remark. This theorem concerns an idealized fixed quadratic with exact curvature matching $S = H$ and separately optimized constant step sizes. It does not identify the neural-network gradient second moment with the Hessian, imply faster convergence at the same learning rate, or cover clipping, normalization, stochastic geometry, or optimizer state. Its role is to characterize the optimization consequence of the spectral transformation under a setting where curvature and gradient geometry coincide.

Proof. Since P is a matrix function of H, the two matrices share an orthonormal eigenbasis. In that basis, PH has positive eigenvalues

$$\nu(\mu) = \mu \sqrt{\frac{\tau}{\mu + \tau}}.$$

The derivative

$$\nu'(\mu) = \frac{\sqrt{\tau}(\mu/2 + \tau)}{(\mu + \tau)^{3/2}} > 0$$

shows that the extreme eigenvalues correspond to $\mu_{\min}$ and $\mu_{\max}$. Their ratio is exactly κ_{opt}. For the error $e_t = \phi_t - \phi_*$, the update is

$$e_{t+1} = (I - \eta P H) e_t.$$

The optimal positive constant step size minimizes $\max_j |1 - \eta \nu_j|$. Equating the two extreme magnitudes gives

$$\eta_* = \frac{2}{\nu_{\max} + \nu_{\min}}, \qquad q_* = \frac{\nu_{\max} - \nu_{\min}}{\nu_{\max} + \nu_{\min}} = \frac{\kappa_{\mathrm{opt}} - 1}{\kappa_{\mathrm{opt}} + 1}.$$

Thus every eigen-coordinate contracts in magnitude by at most q_*. Writing $Q(\phi_t) = \frac{1}{2} \sum_j \mu_j e_{t,j}^2$ gives $Q(\phi_t) \le q_*^{2t} Q(\phi_0)$.

For unpreconditioned GD, the same calculation replaces κ_{opt} by $\kappa(H)$. When $\mu_{\max} > \mu_{\min}$ and τ is finite, $\kappa_{\mathrm{opt}} < \kappa(H)$. The function $(x - 1)/(x + 1)$ is strictly increasing for $x \ge 1$, proving the improved worst-case rate. Finally,

$$\lim_{\tau \downarrow 0} \kappa_{\mathrm{opt}} = \kappa(H) \sqrt{\frac{\mu_{\min}}{\mu_{\max}}} = \sqrt{\kappa(H)}.$$

The step size must be adjusted in this limit because the preconditioner also contracts the overall update scale. □

A.9 PRACTICAL NONNEGATIVE SAMPLE WEIGHTS

The raw coefficients $\widehat{w} = f_\tau(K)m$ need not be elementwise nonnegative, even though $f_\tau(K)$ is positive definite. We therefore train with

$$w = \frac{[\widehat{w}]_+}{Z}, \qquad Z = \mathbf{1}^\top [\widehat{w}]_+.$$

The following result establishes that this normalization is well defined and quantifies the distortion introduced by clipping.

Proposition 8 (Feasibility and effect of clipping)**.** *Let $m \in \Delta_b$ be nonzero and $\tau > 0$. Then $Z > 0$, so $w \in \Delta_b$.*

Let

$$n = [-\widehat{w}]_+, \qquad g_w = A^\top w.$$

Then

$$g_w = \frac{\widehat{g} + A^\top n}{Z},$$

and hence

$$\left\| g_w - \frac{\widehat{g}}{Z} \right\|_2 = \frac{\sqrt{n^\top K n}}{Z}.$$

Moreover,

$$\langle g, g_w \rangle \geq \frac{\langle g, \widehat{g} \rangle - \|g\|_2 \sqrt{n^\top K n}}{Z}.$$

If $\widehat{w} \geq 0$, then $g_w = \widehat{g}/(\mathbf{1}^\top \widehat{w})$, so clipping changes only the scale of the raw direction.

Proof. Since $f_\tau(\lambda) > 0$ for all $\lambda \geq 0$, $f_\tau(K)$ is positive definite. Therefore,

$$m^\top \widehat{w} = m^\top f_\tau(K) m > 0.$$

Because $m \geq 0$, $\widehat{w}$ must have at least one positive coordinate, implying $Z > 0$ and hence $w \in \Delta_b$.

Since $[\widehat{w}]_+ = \widehat{w} + n$,

$$g_w = A^\top w = \frac{A^\top \widehat{w} + A^\top n}{Z} = \frac{\widehat{g} + A^\top n}{Z}.$$

Using $K = AA^\top$,

$$\|A^\top n\|_2^2 = n^\top K n,$$

which gives the stated deviation bound. The alignment bound follows from Cauchy–Schwarz:

$$\langle g, A^\top n \rangle \geq -\|g\|_2 \|A^\top n\|_2.$$

If $\widehat{w} \geq 0$, then $n = 0$, yielding the final claim. □

Thus, the exact spectral guarantees in the main text apply to the raw direction $\widehat{g}$; the practical direction remains close up to normalization whenever the negative part of $\widehat{w}$ is small in the Gram-induced seminorm $\sqrt{n^\top K n}$.

A.10 MEASUREMENT GEOMETRY AND EFFICIENT CONSTRUCTION

We first derive the classifier-head factorization used in equation 3. Let $h_i \in \mathbb{R}^{d_h}$ be the representation of example i, and consider a C-class linear head

$$z_i = W h_i + b, \qquad W \in \mathbb{R}^{C \times d_h}, \qquad b \in \mathbb{R}^C.$$

Let

$$r_i := \nabla_{z_i} \ell_i \in \mathbb{R}^C.$$

For cross-entropy with target distribution $\bar{y}_i$ and $p_i = \mathrm{softmax}(z_i)$, this reduces to $r_i = p_i - \bar{y}_i$. By the chain rule,

$$\nabla_W \ell_i = r_i h_i^\top, \qquad \nabla_b \ell_i = r_i.$$

Introducing the augmented feature

$$\widetilde{h}_i = \begin{bmatrix} h_i \\ 1 \end{bmatrix} \in \mathbb{R}^{d_h+1}$$

and augmented classifier matrix $\widetilde{W} = [W\ b]$, we have $z_i = \widetilde{W}\widetilde{h}_i$ and

$$\nabla_{\widetilde{W}} \ell_i = r_i \widetilde{h}_i^\top.$$

Thus, the vectorized per-example head gradient is

$$a_i^{\text{head}} = \text{vec}\big(\nabla_{\widetilde{W}} \ell_i\big) = \widetilde{h}_i \otimes r_i \in \mathbb{R}^{C(d_h+1)}.$$

Therefore,

$$K_{ij}^{\text{head}} = \langle a_i^{\text{head}}, a_j^{\text{head}} \rangle = \left\langle r_i \widetilde{h}_i^\top, r_j \widetilde{h}_j^\top \right\rangle_F = \langle r_i, r_j \rangle \langle \widetilde{h}_i, \widetilde{h}_j \rangle,$$

which proves equation 3.

Let $R \in \mathbb{R}^{b \times C}$ and $\widetilde{H} \in \mathbb{R}^{b \times (d_h+1)}$ stack $r_i^\top$ and $\widetilde{h}_i^\top$ by row. Applying the preceding identity entrywise gives

$$K^{\text{head}} = (RR^\top) \odot (\widetilde{H}\widetilde{H}^\top), \tag{8}$$

where $\odot$ denotes the Hadamard product. Both factors are PSD, so $K^{\text{head}} \succeq 0$ follows.

A naive construction would materialize the per-example head-gradient matrix $A^{\text{head}} \in \mathbb{R}^{b \times C(d_h+1)}$, requiring $O(bCd_h)$ memory and $O(b^2 Cd_h)$ computation to form its Gram matrix. Equation (8) avoids this joint representation, requiring $O(bC + bd_h + b^2)$ memory and $O\big(b^2(C + d_h)\big)$ computation for the two Gram products.

Remark. When $\phi \subseteq \theta$, the spectral guarantees derived from K apply to the raw direction in the measurement coordinates, rather than to the full-network update.

B Experimental Details and Additional Results

This appendix specifies the evaluation protocol and training configurations, then presents learning curves, sensitivity analyses, mechanism ablations, pretraining results, and computational costs. Each analysis complements the benchmark results in Table 1.

B.1 Evaluation protocol

Metrics and checkpoints. We report accuracy for image classification, sentiment classification, and multiple-choice tasks, and exact match for GSM8K. Results use the held-out splits in Table 2. The checkpoint triplets in Table 1 are measured in epochs, except for MMLU, where they denote optimizer updates. Final performance is measured at the last scheduled checkpoint. AUC denotes the area under the held-out performance curve, normalized by the length of the evaluation window. For MMLU, we evaluate every reported checkpoint on the official test set of $14{,}042$ examples and compute AUC from the resulting test-accuracy curve. For image classification, we compute AUC over the first 10% of optimization steps; for language fine-tuning, where the training horizon is substantially shorter, we compute AUC over the full training trajectory. Percentage-point (pp) differences are computed by subtracting the reported mean accuracies or exact-match scores.

Randomness and comparisons. Unless stated otherwise, entries are means and sample standard deviations over three seeds. Across-seed standard deviations describe variability in each method's results. Improved settings therefore refer to comparisons of reported means.

B.2 Datasets and implementation

Table 2 summarizes the data splits, models, and training budgets. The vision experiments use CIFAR-10/100 (Krizhevsky, 2009) and ImageNet-1K (Deng et al., 2009), with ResNet-18/50 (He et al., 2016) and CCT-7 (Hassani et al., 2021). Language fine-tuning uses DeBERTa-v3-base (He et al., 2023) on SST-2 (Socher et al., 2013), and Qwen3 Base models (Yang et al., 2025) on GSM8K (Cobbe et al., 2021), CommonsenseQA (Talmor et al., 2019), and MMLU (Hendrycks et al., 2021).

Dataset	Train	Evaluation	Model	Epochs
CIFAR-10	50,000	10,000 test	ResNet-18 / CCT-7	200 / 300
CIFAR-100	50,000	10,000 test	ResNet-18	200
ImageNet-1K	1,281,167	50,000 val.	ResNet-50	90
SST-2	67,349	872 val.	DeBERTa-v3-base	3
GSM8K	7,473	1,319 test	Qwen3-0.6B + LoRA	3
CommonsenseQA	9,741	1,221 val.	Qwen3-0.6B + LoRA	3
MMLU	10,000	14,042 test	Qwen3-1.7B + LoRA	1

Table 2: Datasets, evaluation splits, and training budgets for Table 1. Qwen3 models use the Base checkpoints with LoRA. MMLU uses a $10{,}000$-example training subset and runs for 313 optimizer updates; the other checkpoint triplets in the main table are measured in epochs.

Optimizers and comparators. Base denotes training with the listed optimizer and the reference example weights. We evaluate SGD, AdamW (Loshchilov & Hutter, 2019), and Muon (Jordan et al., 2024); Table 1 additionally reports Aligned-MTL. Within each task, the architecture, data pipeline, augmentation, and optimization schedule are held fixed between Base and LISC. Their selected weight decay differs in three settings, as detailed below.

LISC implementation. Weights are recomputed for each mini-batch using $\gamma = 0.03$, except in the sensitivity study below. Unless stated otherwise, we clip the raw LISC coefficients at zero and renormalize them to sum to one. Vision experiments use classifier-head gradients to construct the Gram matrix by default, as described in Section 4.2. Qwen3 models are fine-tuned using rank-16 LoRA (Hu et al., 2022) on the major linear projections of each Transformer block, with embeddings and the language-model head frozen; LISC uses gradients with respect to all trainable adapters. The

resulting sample weights are detached before differentiating the weighted loss with respect to all trainable parameters.

Aligned-MTL implementation. We adapt Aligned-MTL (Senushkin et al., 2023) to sample aggregation by treating each example, or sequence in language modeling, as an individual objective. We use the same reference weights and measurement coordinates as LISC: the classifier head for vision and SST-2, and all trainable LoRA parameters for Qwen3. Its undamped inverse-square-root transformation equalizes the retained singular values of the gradient matrix at the smallest retained value.

Coefficients are recomputed for each physical mini-batch and held fixed when differentiating the weighted loss with respect to all trainable parameters. We preserve their signs and scale, without clipping, normalization, or additional norm matching. Gradients are accumulated where required before applying the task-specific optimizer and schedule.

For numerical stability, we retain Gram eigenvalues satisfying $\lambda_j > \max(10^{-12}, 10^{-6}\lambda_{\max}(K))$ and use the reference weights if none are retained.

B.3 Training configurations and hyperparameter search

Tables 3 and 4 list the Base and LISC configurations associated with the reported results. Within each setting we keep architecture, data pipeline, augmentation, and training budget fixed across methods. When hyperparameters are swept, methods use the same candidate grid and are selected independently for each method according to the best mean final-checkpoint accuracy over three seeds. The selected batch size, learning rate, and schedule agree within each Base–LISC pair, whereas weight decay differs for CIFAR-10 with SGD, ImageNet-1K with SGD, and GSM8K. Accordingly, the main table compares the reported training configurations; a fixed-configuration comparison is provided by the hyperparameter sensitivity analysis in Appendix B.7.

Table 3: Vision configurations for Base and LISC. B is the global batch size and η the peak learning rate. Weight decay is listed separately for the two methods. RN18/50 denotes ResNet-18/50.

Dataset/model	Optimizer	B	η	WD (Base)	WD (LISC)	Schedule
CIFAR-10 / RN18	SGD	128	0.1	5×10^{-4}	3×10^{-4}	Cosine
CIFAR-10 / RN18	Muon	128	0.16	10^{-2}	10^{-2}	Cosine
CIFAR-10 / CCT-7	AdamW	128	5.5×10^{-4}	0.05	0.05	10-epoch warm-up, cosine
CIFAR-100 / RN18	SGD	128	0.1	5×10^{-4}	5×10^{-4}	Cosine
CIFAR-100 / RN18	Muon	128	0.16	10^{-2}	10^{-2}	Cosine
ImageNet-1K / RN50	SGD	1,024	0.4	10^{-4}	7.5×10^{-5}	Warm-up; drops at 30, 60, 80
ImageNet-1K / RN50	Muon	1,024	0.02	10^{-2}	10^{-2}	Warm-up; drops at 30, 60, 80

Table 4: Language fine-tuning configurations for Base and LISC with AdamW. $B_{\mathrm{p}}/B_{\mathrm{eff}}$ denotes physical/effective batch size; η is the peak learning rate. All schedules use linear decay after warm-up.

Dataset/model	$B_{\mathrm{p}}/B_{\mathrm{eff}}$	η	WD (Base)	WD (LISC)	Warm-up
SST-2 / DeBERTa-v3-base	$32/32$	2×10^{-5}	10^{-2}	10^{-2}	6%
GSM8K / Qwen3-0.6B	$4/32$	2×10^{-4}	2×10^{-3}	0	3%
CommonsenseQA / Qwen3-0.6B	$8/32$	4×10^{-4}	10^{-2}	10^{-2}	3%
MMLU / Qwen3-1.7B	$8/32$	2×10^{-4}	2×10^{-3}	2×10^{-3}	3%

Table 5 gives the specified search grids. For vision, the learning rate scales linearly with batch size. For GSM8K, physical batch size varies while effective batch size remains 32. This distinction matters for LISC because the physical mini-batch determines the examples entering each Gram matrix.

Sweep	Batch	Learning rate	Weight decay
CIFAR-100 / SGD	$\{128, 256, 512\}$	$\{0.05, 0.1\}b/128$	$\{10^{-4}, 3\times10^{-4}, 5\times10^{-4}\}$
CIFAR-100 / Muon	$\{128, 256, 512\}$	$\{0.08, 0.16\}b/128$	$\{0, 10^{-4}, 10^{-2}\}$
CIFAR-10 / CCT-7	$\{128, 256, 512\}$	$\{2.75, 5.5\}\times10^{-4}b/128$	$\{10^{-4}, 10^{-2}, 0.05\}$
GSM8K / AdamW	$\{4, 8\}$	2×10^{-4}	$\{0, 4\times10^{-4}, 2\times10^{-3}, 10^{-2}\}$
MMLU / AdamW	$\{4, 8\}$	$\{1, 2, 4\} \times 10^{-4}$	$\{2\times10^{-3}, 10^{-2}\}$

Table 5: Hyperparameter grids for the sensitivity experiments. In the vision grids, b is the batch size and the learning rate scales with $b/128$. GSM8K varies physical batch size while retaining effective batch size 32. These grids are distinct from the selected configurations in Tables 3 and 4.

B.4 Choice of γ for LISC

To evaluate sensitivity to the damping parameter γ, we train a ResNet-18 on CIFAR-10 (batch size 128) with LISC using classifier-head measurement coordinates across ten values of $\gamma \in [0.01, 0.99]$. As shown in Table 6, final accuracy is remarkably stable (roughly 95.3%–95.4%). However, smaller γ values enforce stronger spectral correction, which accelerates early-stage learning. Specifically, $\gamma = 0.01$ maximizes accuracy at Epoch 5, while $\gamma = 0.03$ yields the highest AUC through Epoch 20. Because $\gamma = 0.03$ provides optimal early acceleration without sacrificing final performance, we adopt it for all subsequent experiments.

Table 6: Sensitivity to γ on CIFAR-10 with ResNet-18, SGD, classifier-head geometry, and batch size 128. Entries are mean $\pm$ sample standard deviation of accuracy (%) over three seeds. Final denotes epoch 200.

γ	Epoch 5	Epoch 20	AUC (0–20)	Final
0.01	$\mathbf{76.85 \pm 2.78}$	85.28 ± 1.47	75.31 ± 0.92	95.29 ± 0.03
0.03	76.48 ± 2.80	$\mathbf{85.36 \pm 1.63}$	$\mathbf{75.48 \pm 1.04}$	95.37 ± 0.12
0.05	76.69 ± 2.87	85.29 ± 2.65	75.29 ± 1.06	95.35 ± 0.20
0.10	76.44 ± 2.21	85.02 ± 0.42	74.12 ± 1.04	95.37 ± 0.06
0.20	75.59 ± 1.97	84.53 ± 3.39	74.06 ± 1.45	$\mathbf{95.43 \pm 0.18}$
0.30	74.45 ± 1.99	84.63 ± 0.52	73.52 ± 1.61	95.38 ± 0.10
0.40	76.43 ± 2.44	84.80 ± 2.06	74.35 ± 1.25	95.39 ± 0.15
0.60	74.59 ± 1.71	84.56 ± 1.01	72.99 ± 1.42	95.37 ± 0.20
0.80	74.75 ± 2.44	84.79 ± 1.47	73.98 ± 1.80	95.40 ± 0.18
0.99	75.87 ± 2.44	84.55 ± 0.89	73.44 ± 1.50	95.34 ± 0.19

B.5 Additional learning curves and training behavior

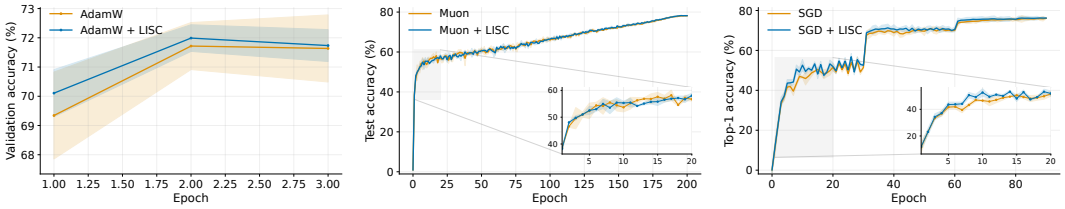
Figure 5 complements the main learning-curve figure with the additional settings. With the exception of CIFAR-10 with CCT, the early gain persists with LISC till the final stage of training. The difference is more pronounced on heterogeneous data, where LISC accelerates early learning and retains this advantage till the final stage.

B.6 Language-model pretraining

We pretrain GPT-2 Small (Radford et al., 2019) on FineWeb (Penedo et al., 2024), using Muon for matrix-valued hidden-layer parameters and AdamW for the remaining parameters. LISC measures example interactions using the final MLP output projection. Table 7 reports held-out loss at five checkpoints through 3,000 updates. Mean loss is lower with LISC at every checkpoint, with reductions ranging from 0.0002 to 0.0028. These differences are small relative to the reported across-seed variation; paired results are needed to characterize their uncertainty.

B.7 Sensitivity across hyperparameter configurations

Figure 6 reports LISC minus Base accuracy for paired hyperparameter configurations across different settings: 18 each for CIFAR-100 with SGD, CIFAR-100 with Muon, and CIFAR-10 with CCT-7,



(a) CommonsenseQA, Qwen3-0.6B (b) CIFAR-100, ResNet-18, Muon (c) Imagenet, ResNet-50, SGD

(d) CIFAR-10, CCT-7, AdamW (e) ImageNet, ResNet-50, Muon (f) Train Accuracy, CIFAR-10, Muon

Figure 5: **Additional learning curves across vision and language settings** (a–e). Held-out performance across various settings; (f) Training Accuracy on CIFAR-10 with ResNet-18 and Muon.

Table 7: **GPT-2 Small pretraining on FineWeb.** Held-out validation loss and perplexity; lower is better for both. Results reported as mean ± sample standard deviation over three seeds.

	Loss ↓		Perplexity ↓	
Updates	Muon	Muon + LISC	Muon	Muon + LISC
50	7.1625 ± 0.0049	$\mathbf{7.1598 \pm 0.0063}$	1290.084 ± 6.364	$\mathbf{1286.618 \pm 8.106}$
100	6.6265 ± 0.0028	$\mathbf{6.6237 \pm 0.0045}$	754.855 ± 2.138	$\mathbf{752.740 \pm 3.362}$
1,000	4.8330 ± 0.0058	$\mathbf{4.8328 \pm 0.0073}$	125.590 ± 0.728	$\mathbf{125.565 \pm 0.915}$
2,000	4.4199 ± 0.0028	$\mathbf{4.4178 \pm 0.0031}$	83.087 ± 0.231	$\mathbf{82.911 \pm 0.256}$
3,000	4.1864 ± 0.0030	$\mathbf{4.1849 \pm 0.0030}$	65.784 ± 0.196	$\mathbf{65.689 \pm 0.199}$

and 8 for GSM8K with Qwen3-0.6B. Each vision grid crosses three batch sizes, two learning rates, and three weight-decay values from Table 5. Comparing methods within the same cell holds these hyperparameters fixed and complements the selected-configuration results in the main table. The reported pattern indicates widespread early gains for SGD and CCT-7, while final gains vary across configurations.

B.8 Empirical properties of the LISC transformation

The main text establishes several structural properties of the raw LISC direction $\widehat{g}$: it remains positively aligned with the reference mini-batch gradient g (Proposition 4), clipping is introduced only to obtain a valid nonnegative weighting of examples (Section 4), and the inverse-square-root filter improves the condition number of the corresponding *effective* sample-space operator (Section A.7). Here we examine these properties empirically on the classifier-head geometry used in our vision experiments.

Alignment with the mini-batch gradient. Proposition 4 shows that the raw LISC direction $\widehat{g}$ is positively aligned with the ordinary mini-batch gradient g and has smaller norm. Figure 7a provides the corresponding empirical picture. The cosine $\cos(g, \widehat{g})$ remains strictly positive throughout training and is typically high. In particular, the full-parameter and classifier-head variants are nearly indistinguishable during the early and middle phases of training, with cosine values above 0.85 in our runs. Later in training, the classifier-head cosine decreases, reaching approximately 0.57, but it remains positive throughout. Thus, in the regimes where LISC produces its strongest acceleration, the practical classifier-head approximation remains closely aligned with the full-parameter transformation.

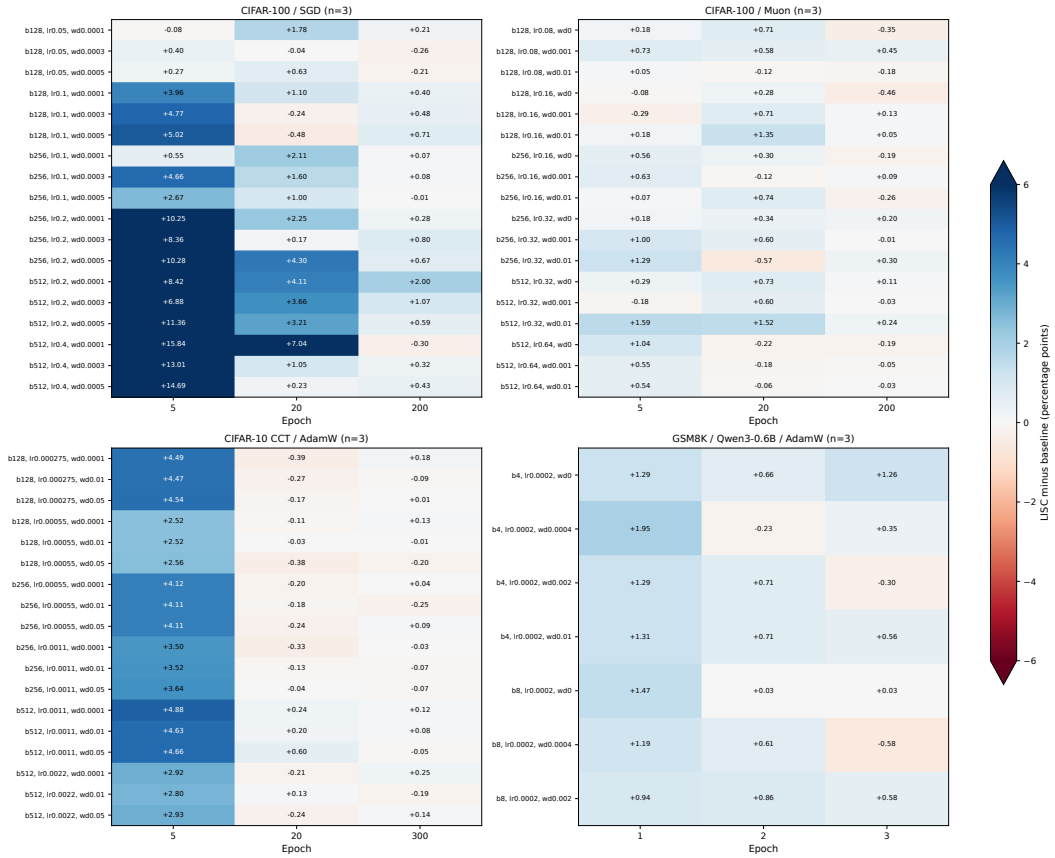
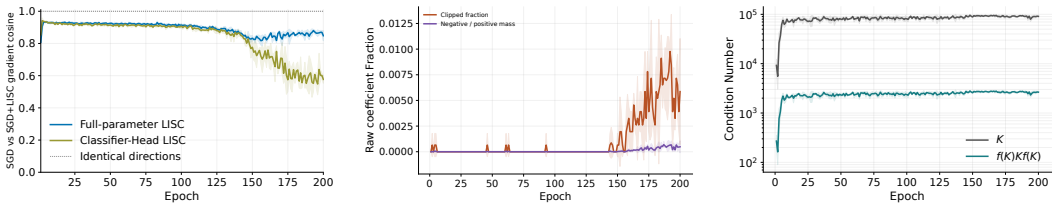


Figure 6: **Hyperparameter sensitivity.** LISC minus Base mean accuracy (pp) for paired hyperparameter configurations, averaged over three matched seeds per cell. Color saturates beyond ± 6 pp; printed values retain the full difference. Positive values favor LISC. Grid cells share datasets and recipes and should not be treated as independent replications.



(a) Gradient cosine $\cos(g, \widehat{g})$.

(b) Fraction of non-positive raw coefficients.

(c) Condition number of the effective operator.

Figure 7: **Empirical properties of the LISC transformation. Left:** The LISC direction remains positively aligned with the reference mini-batch gradient throughout training, as proposed in Proposition 4. **Middle:** Only a very small fraction of raw coefficients are non-positive. The clipped fraction remains below 1% at all times. **Right:** The inverse-square-root filter improves the condition number of the *effective* sample-space operator on the retained modes.

The positive cosine also clarifies the implication of Proposition 4: LISC does not reverse the update direction, but instead redistributes its energy while preserving descent on the local reference objective.

Effect of clipping. As discussed in Section 4, the raw coefficient vector $\widehat{w} = f_\tau(K)m$ need not be elementwise nonnegative, so we clip negative entries and renormalize before forming the practical weights. Figure 7b shows that this correction is mild in practice. The fraction of non-positive raw coefficients stays close to zero until roughly the middle of training, and even at its maximum remains below 1%. Thus, the empirical behavior of LISC is dominated by the inverse spectral reweighting itself, while clipping introduces only a small perturbation to the raw transformation. We use clipping just as a safeguard ensuring that the practical objective does not explicitly assign negative weight to training examples.

Conditioning of the effective sample-space operator. The theoretical conditioning statement applies to the *raw* filtered operator induced by the inverse-square-root response, not to the original Gram matrix K itself. Specifically, if the positive spectrum of K is retained on a fixed set of modes, then the raw filter changes the effective sample-space operator from K to $f_\tau(K)\, K\, f_\tau(K)$, whose condition number satisfies $\kappa_{\text{eff}} = \kappa(K)\frac{\lambda_{\min}+\tau}{\lambda_{\max}+\tau} \le \kappa(K)$, with strict improvement whenever the retained eigenvalues are not all equal; see Section A.7. Figure 7c evaluates this effect empirically using classifier-head operators and restricting attention to modes with $\lambda_j \ge 10^{-5}\lambda_{\max}$. The figure shows a consistent reduction in condition number under the raw inverse-square-root filter. Because the practical clipped transformation is nonlinear, it does not admit a single corresponding condition number, so the plot is reported for the raw filtered operator. Figure 7b shows that only tiny fractions of weights are clipped, so its effects should not be large. We emphasize that these are condition numbers of the classifier-head sample-gradient operators, not of the loss Hessian or any curvature matrix.

Taken together, Figure 7 supports the interpretation developed in the main text. In the regimes where LISC is most effective, the practical classifier-head implementation remains strongly aligned with the full-parameter direction, clipping affects only a negligible fraction of examples, and the raw inverse-square-root filter measurably improves the conditioning of the effective sample-space operator. These observations are consistent with the view that LISC works primarily by rebalancing directional influence in the mini-batch gradient rather than by introducing a qualitatively different optimization objective.

B.9 Runtime and Memory

We measure the computational overhead introduced by constructing the measurement geometry and computing LISC weights at each optimization step. All measurements are obtained from single-GPU runs on an NVIDIA A100-SXM4-40GB. Wall-clock time includes the complete training run, and memory is measured as peak allocated CUDA memory. Results are summarized in Table 8.

Table 8: **Computational overhead of LISC.** Wall-clock training time is reported in minutes, and memory denotes peak allocated CUDA memory in GiB.

Dataset / model	Optimizer	Base (min)	+ LISC (min)	Overhead	Memory (Base / LISC)
CIFAR-10 / ResNet-18	SGD	24.26 ± 0.68	29.26 ± 1.01	20.6%	0.590 / 0.587
CIFAR-10 / ResNet-18	Muon	69.44 ± 1.08	74.41 ± 3.17	7.2%	0.487 / 0.485
CIFAR-10 / CCT-7	AdamW	103.26 ± 0.66	110.58 ± 0.53	7.1%	3.673 / 3.671
CIFAR-100 / ResNet-18	SGD	23.44 ± 0.62	28.44 ± 0.46	21.4%	0.491 / 0.488
CIFAR-100 / ResNet-18	Muon	65.60 ± 1.04	70.57 ± 0.99	7.6%	0.491 / 0.489
ImageNet / ResNet-50	SGD	962.22 ± 10.80	1354.33 ± 11.16	40.8%	12.025 / 14.194
GSM8K / Qwen3-0.6B	AdamW	128.22 ± 5.38	183.81 ± 1.84	43.4%	13.430 / 13.430
CommonsenseQA	AdamW	36.49 ± 0.12	51.53 ± 0.17	41.2%	19.488 / 19.488
MMLU / Qwen3-1.7B	AdamW	32.11 ± 0.03	44.52 ± 0.19	38.6%	23.361 / 23.946
GPT-2 Small / FineWeb	Muon	30.05 ± 0.30	34.12 ± 0.20	13.6%	7.098 / 7.147

The runtime overhead varies across datasets. LISC incurs approximately 7–21% runtime overhead on CIFAR, 14% on GPT-2 pretraining, and 39–43% on ImageNet and the measured language fine-tuning tasks. Peak allocated memory changes little in most settings, with the largest increase on ImageNet. The reported acceleration concerns improved generalization per optimization step and does not imply a corresponding wall-clock speedup.

C COMPLEMENTARITY WITH OPTIMIZER-SIDE TRANSFORMATIONS: A SEPARATION FOR MUON

LISC modifies the coefficients used to aggregate per-example gradients before the resulting gradient is passed to an optimizer. This suggests that LISC can affect aspects of the update that are not accessible to transformations applied only after gradient aggregation. We make this distinction explicit for Muon (Jordan et al., 2024; Liu et al., 2025) using a simple two-feature construction.

The purpose of this example is not to establish that LISC must improve Muon on every problem. Rather, we exhibit a setting in which the two operations act on different degrees of freedom. Muon transforms the singular values of an aggregated matrix while preserving its singular-vector subspaces, whereas LISC can change the aggregated matrix itself through its sample coefficients. Thus, it establishes that optimizer-side spectral conditioning does not subsume sample-side spectral reweighting, and that composing the two can yield behavior unavailable to Muon alone.

C.1 A TWO-DIRECTION MODEL

Consider binary classification with $y \in \{-1, +1\}$ and

$$x = y \begin{bmatrix} \alpha s \\ \beta t \end{bmatrix}, \qquad \alpha > \beta > 0,$$

where $s, t \in \{-1, +1\}$. We interpret s as a training-specific feature and t as a feature whose correlation with the label persists at test time. Let the training distribution be

$$p_{\mathrm{tr}}(s, t) = \frac{1}{4}\left(1 + \delta_s s + \delta_t t\right), \tag{9}$$

where $\delta_s, \delta_t > 0$ and $\delta_s + \delta_t \leq 1$. Then

$$\mathbb{E}_{\mathrm{tr}}[s] = \delta_s, \qquad \mathbb{E}_{\mathrm{tr}}[t] = \delta_t, \qquad \mathbb{E}_{\mathrm{tr}}[st] = 0.$$

At test time, suppose only the s correlation disappears:

$$p_{\mathrm{te}}(s, t) = \frac{1}{4}(1 + \delta_t t). \tag{10}$$

Consequently,

$$\mathbb{E}_{\mathrm{tr}}[yx] = \begin{bmatrix} \alpha\delta_s \\ \beta\delta_t \end{bmatrix}, \qquad \mathbb{E}_{\mathrm{te}}[yx] = \begin{bmatrix} 0 \\ \beta\delta_t \end{bmatrix}.$$

Thus, the first coordinate contributes predictive signal during training but not under the shifted test distribution.

To obtain a matrix-valued parameter to which Muon applies naturally, consider

$$f_W(x) = v^\top W x, \qquad W \in \mathbb{R}^{p\times 2}, \qquad \|v\|_2 = 1,$$

with fixed $v \in \mathbb{R}^p$. Statistically, this is an ordinary two-dimensional linear classifier with effective parameter $\theta = W^\top v$. We use the logistic loss

$$\ell_i(W) = \log\left(1 + \exp(-y_i f_W(x_i))\right).$$

At $W = 0$,

$$G_i := \nabla_W \ell_i(0) = -\frac{1}{2} v q_i^\top, \qquad q_i = \begin{bmatrix} \alpha s_i \\ \beta t_i \end{bmatrix}. \tag{11}$$

Consider a finite mini-batch whose empirical moments equal those in equation 9, i.e.,

$$\frac{1}{b}\sum_i s_i = \delta_s, \qquad \frac{1}{b}\sum_i t_i = \delta_t, \qquad \frac{1}{b}\sum_i s_i t_i = 0. \tag{12}$$

With uniform reference weights $m_i = 1/b$, the ordinary aggregated gradient is

$$G = \frac{1}{b}\sum_i G_i = -\frac{1}{2} v \mu^\top, \qquad \mu = \begin{bmatrix} \alpha\delta_s \\ \beta\delta_t \end{bmatrix}.$$

In particular, G has rank one, and its right singular direction is $\mu/\|\mu\|_2$.

C.2 MUON CANNOT CHANGE THE RANK-ONE FEATURE DIRECTION

Muon applies an approximate matrix orthogonalization to the momentum matrix. The relevant observation here is that Newton–Schulz-type spectral transformations preserve singular vectors.

Lemma 9 (Rank-one invariance under Muon)**.** *Let*

$$G = c\, v\mu^\top$$

be rank one. Any transformation obtained from iterations whose terms are of the form

$$(GG^\top)^k G$$

maps G to a scalar multiple of G. Consequently, the finite Newton–Schulz transformation used by Muon cannot rotate the right singular direction $\mu/\|\mu\|_2$ of this rank-one input.

Proof. Since $\|v\|_2 = 1$,

$$GG^\top = c^2\|\mu\|_2^2 vv^\top.$$

Hence, for every integer $k \geq 0$,

$$(GG^\top)^k G = \left(c^2\|\mu\|_2^2\right)^k G.$$

Every polynomial combination of these terms is therefore a scalar multiple of G, and its left and right singular directions are unchanged. □

At the first optimization step, Muon's momentum is itself a scalar multiple of G, so Lemma 9 applies. More generally, the lemma isolates the relevant algebraic property: once the samples have produced this rank-one aggregate, an optimizer-side spectral transformation can modify its scale but not its feature-space direction.

C.3 LISC CHANGES THE DIRECTION BEFORE MUON ACTS

We next compute the LISC transformation for the same batch. Let $a_i = \mathrm{vec}(G_i)$. By equation 11,

$$K_{ij} = a_i^\top a_j = \frac{1}{4} q_i^\top q_j = \frac{1}{4}\left(\alpha^2 s_i s_j + \beta^2 t_i t_j\right).$$

Writing

$$s = (s_1, \ldots, s_b)^\top, \qquad t = (t_1, \ldots, t_b)^\top,$$

we therefore have

$$K = \frac{\alpha^2}{4} ss^\top + \frac{\beta^2}{4} tt^\top.$$

The moment condition in equation 12 implies $s^\top t = 0$, while $\|s\|_2^2 = \|t\|_2^2 = b$. Thus, K has two nonzero eigenvectors

$$u_s = \frac{s}{\sqrt{b}}, \qquad u_t = \frac{t}{\sqrt{b}},$$

with eigenvalues

$$\lambda_s = \frac{b\alpha^2}{4}, \qquad \lambda_t = \frac{b\beta^2}{4}.$$

Because $\alpha > \beta$, the training-specific s direction is the dominant Gram mode.

For $m = \mathbf{1}/b$, its overlaps with these modes are

$$u_s^\top m = \frac{\delta_s}{\sqrt{b}}, \qquad u_t^\top m = \frac{\delta_t}{\sqrt{b}}.$$

Choose $\tau = \gamma\lambda_s$, as in the main method. The LISC gains along the two modes are

$$f_s = \sqrt{\frac{\gamma}{1+\gamma}}, \qquad f_t = \sqrt{\frac{\gamma\alpha^2}{\beta^2 + \gamma\alpha^2}}.$$

Since $\alpha > \beta$, we have $f_s < f_t$.

The raw LISC-weighted gradient is consequently

$$\widehat{G} = -\frac{1}{2} v \widehat{\mu}^\top, \qquad \widehat{\mu} = \begin{bmatrix} f_s \alpha \delta_s \\ f_t \beta \delta_t \end{bmatrix}. \tag{13}$$

Define the ratio between training-specific and stable components by

$$R_{\mathrm{ERM}} = \frac{\alpha \delta_s}{\beta \delta_t}, \qquad R_{\mathrm{LISC}} = \frac{f_s \alpha \delta_s}{f_t \beta \delta_t}.$$

Then

$$\frac{R_{\mathrm{LISC}}}{R_{\mathrm{ERM}}} = \frac{f_s}{f_t} = \sqrt{\frac{\gamma + (\beta/\alpha)^2}{1+\gamma}} < 1. \tag{14}$$

Thus, LISC rotates the aggregated direction away from the higher-energy training-specific coordinate and toward the weaker stable coordinate.

Importantly, LISC has not been given information identifying either feature as "stable" or "spurious." The rotation follows from the larger sample-gradient eigenvalue associated with the s-mode.

C.4 The practical nonnegative weights can realize the same direction

The raw coefficients are also available in closed form. Decomposing m into the two nonzero eigenspaces and the nullspace of K gives

$$\widehat{w}_i = \frac{1}{b}\left[1 - (1-f_s)\delta_s s_i - (1-f_t)\delta_t t_i\right].$$

A sufficient condition for all coefficients to remain positive is

$$1 - (1-f_s)\delta_s - (1-f_t)\delta_t > 0. \tag{15}$$

Under equation 15, the clipping stage of practical LISC is inactive, and normalization multiplies all coefficients by a common positive scalar. Hence the practical LISC gradient has the same direction as $\widehat{G}$ in equation 13.

Muon therefore receives two different rank-one matrices:

$$\text{Muon:} \quad G = -\frac{1}{2} v \mu^\top, \qquad \text{LISC+Muon:} \quad \widehat{G} = -\frac{1}{2} v \widehat{\mu}^\top.$$

By Lemma 9, Muon preserves the respective right singular directions. LISC therefore changes which direction is presented to Muon rather than duplicating Muon's transformation.

C.5 A strict first-order test-loss separation

We finally show that this directional change can improve early test behavior under the distribution shift in equation 10. Consider equal-norm Muon steps and write the resulting effective linear predictor as

$$\theta = \eta u, \qquad \|u\|_2 = 1.$$

Around initialization, the population test logistic loss satisfies

$$\begin{aligned} L_{\mathrm{te}}(\eta u) &= \log 2 - \frac{\eta}{2} \mathbb{E}_{\mathrm{te}}[yx]^\top u + O(\eta^2) \\ &= \log 2 - \frac{\eta}{2} \beta \delta_t u_2 + O(\eta^2). \end{aligned} \tag{16}$$

Thus, among equal-norm steps, the first-order test improvement is determined by the component assigned to the stable coordinate.

For Muon alone,

$$u_{\mathrm{M},2} = \frac{1}{\sqrt{1 + R_{\mathrm{ERM}}^2}},$$

while for LISC followed by Muon,

$$u_{\mathrm{L+M},2} = \frac{1}{\sqrt{1 + R_{\mathrm{LISC}}^2}}.$$

Equation (14) therefore implies

$$u_{\mathrm{L+M},2} > u_{\mathrm{M},2}.$$

Proposition 10 (LISC can strictly complement Muon)**.** *Under the construction above, assume $\alpha > \beta > 0$, $\delta_s, \delta_t > 0$, and the positivity condition equation 15. At initialization, LISC followed by an equal-norm Muon update has a strictly larger first-order decrease in the shifted population test loss than Muon applied to the uniformly aggregated gradient:*

$$\frac{d}{d\eta} L_{\mathrm{te}}(\eta u_{\mathrm{L+M}})\Big|_{\eta=0} < \frac{d}{d\eta} L_{\mathrm{te}}(\eta u_{\mathrm{M}})\Big|_{\eta=0}.$$

Proof. From equation 14, $R_{\mathrm{LISC}} < R_{\mathrm{ERM}}$, so $u_{\mathrm{L+M},2} > u_{\mathrm{M},2}$. Differentiating equation 16 at $\eta = 0$ yields

$$\frac{d}{d\eta} L_{\mathrm{te}}(\eta u)\Big|_{\eta=0} = -\frac{1}{2}\beta\delta_t u_2.$$

Since $\beta\delta_t > 0$, the claimed strict inequality follows. □

This example isolates the complementarity between the two mechanisms. Muon can reshape the matrix produced by a given sample aggregation, but in this rank-one setting it cannot alter the feature direction selected by that aggregation. LISC instead changes the sample coefficients before aggregation, thereby presenting Muon with a different direction. When the dominant training mode does not persist at test time, this difference is sufficient to produce a strict first-order test-loss advantage for LISC+Muon.

D Relationship to Aligned-MTL and Jacobian Descent

Aligned-MTL. Aligned-MTL (Senushkin et al., 2023) was developed for multi-task optimization, with evaluations including scene understanding and reinforcement learning, whereas LISC targets example weighting for early-training performance and generalization in supervised learning. Mathematically, Aligned-MTL equalizes the nonzero singular values of the gradient matrix; LISC instead uses a damped inverse-square-root filter that compresses spectral differences without fully equalizing them. We establish the exact raw-direction equivalence $\widehat{g} = (I + S/\tau)^{-1/2} g$, where $S = A^\top A$ is the local gradient second moment in the measurement coordinates. This characterizes LISC as regularized fractional preconditioning realized through sample coefficients. Aligned-MTL also admits a parameter-space interpretation, so the distinction is the particular damped operator and its conversion into nonnegative, normalized example weights. At finite damping, the resulting raw direction generally differs from Aligned-MTL beyond a scalar rescaling.

Jacobian descent. Jacobian descent (Quinton & Rey, 2024) provides a general framework encompassing per-example objectives and Gram-based aggregation; LISC specifies a particular weighting rule within this broader framework. Its UPGrad aggregator seeks a direction that does not conflict with any individual gradient, whereas LISC rebalances spectral influence without requiring every example's loss to decrease. LISC can therefore alter the aggregate even in the absence of gradient conflict. Computational cost is also relevant: our evaluated Jacobian-descent implementation required approximately 11 hours for a CIFAR-100 run under our experimental setup. This is an implementation- and budget-dependent measurement, rather than an intrinsic cost of the framework. For classification, LISC constructs its measurement Gram matrix directly from classifier-head features and loss derivatives, avoiding full per-example parameter gradients. Our emphasis is thus on a specific spectral rule, a practical measurement strategy, and their effects on early-training performance and generalization.