Symmetry-Compatible Matrix-Gradient Methods: Equivariant Updates, Spectral Operators, and Convergence

Tim Tsz-Kit Lau* Weijie Su*

July 26, 2026

Abstract

We develop a symmetry-compatible framework for first-order methods on matrix optimization problems. The central principle is that the update rule for a matrix variable should be equivariant with respect to the natural symmetry group acting on that variable. For matrix representations of linear operators, this leads to bi-orthogonal equivariance under left and right orthogonal changes of basis. We show that this coordinate-free consistency requirement characterizes spectral matrix-gradient methods, whose update maps preserve singular-vector structure and act only on singular values. This viewpoint gives a unified interpretation of stochastic spectral descent, Muon, Scion, and polar gradient methods. We then extend the same principle to matrix variables with one-sided, discrete, or affine symmetry structures, leading to right-spectral, row-normalized, centered, and hybrid row-norm/spectral updates. We establish descent and convergence guarantees under smoothness, alignment, and Polyak–Łojasiewicz conditions, including variants with momentum and inexact spectral or polar oracles. As a motivating application, we instantiate the framework for matrix-valued parameters in modern neural network architectures. The results suggest that equivariance provides a unifying principle for designing geometry-aware matrix-gradient methods.

Key words. Matrix optimization, equivariant matrix-gradient methods, spectral operators, symmetry-compatible optimization methods, first-order methods

MSC codes. 90C26, 90C25, 90C30, 65F25, 65F08, 68T07

1 Introduction

Matrix-valued variables arise throughout continuous optimization, numerical linear algebra, statistics, signal processing, control, and machine learning. Although a matrix can be vectorized by choosing an ordering of its entries, vectorization often obscures the algebraic and geometric structure that distinguishes matrix optimization from ordinary vector optimization: singular values, invariant subspaces, ranks, left and right orthogonal actions, and, in many applications, one-sided, discrete, or affine symmetries. These structures play a central role in matrix analysis and spectral optimization, but they are not automatically respected by generic coordinatewise first-order methods, including the coordinatewise adaptive gradient methods widely used in deep learning, such as Adam [33], Adafactor [60], RMSprop [67], and AdaGrad [16, 50].

This paper develops a symmetry-compatible framework for first-order optimization with matrix-valued variables. The guiding principle is that the update rule for a matrix variable should be equivariant with respect to the natural symmetry group acting on that variable. Equivariance

*University of Pennsylvania, Philadelphia, PA 19104, USA. Emails: timlautk@gmail.com, suw@wharton.upenn.edu.

means that if a matrix direction is represented after a change of coordinates, relabeling, or quotient-preserving transformation, then the optimizer update transforms in the corresponding way. Thus, the optimizer is not merely a rule on arrays of entries, but a coordinate-consistent rule on the underlying matrix object.

For a matrix representing a linear operator between Euclidean spaces, the natural transformations are left and right orthogonal changes of basis. This leads to bi-orthogonal equivariance: if an update direction $D \in \mathbb{R}^{m\times n}$ is transformed to $PDQ^\top$, where $P \in \mathbb{O}^m$ and $Q \in \mathbb{O}^n$, then the update map $\mathcal{U}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ should satisfy

$$\mathcal{U}(PDQ^\top) = P\mathcal{U}(D)Q^\top.$$

We show that this coordinate-free consistency requirement leads exactly to spectral update maps, which preserve singular-vector structure and act only on singular values:

$$D = U\,\mathrm{Diag}_{m,n}(\sigma(D))V^\top \qquad \Longrightarrow \qquad \mathcal{U}(D) = U\,\mathrm{Diag}_{m,n}(\psi(\sigma(D)))V^\top,$$

for a suitable singular-value map ψ. From this viewpoint, stochastic spectral descent [6–8], Muon [31], Scion [56], and polar gradient methods [36] are members of a common family of bi-orthogonally equivariant matrix-gradient methods, differing mainly in their singular-value map, normalization, scaling, and numerical oracle.

The same principle applies beyond full bi-orthogonal symmetry. Many matrix variables have distinguished row or column structures, so their natural symmetries are one-sided, discrete, or affine rather than fully orthogonal. For example, if rows index categories, tokens, groups, or experts, arbitrary row rotations are not meaningful, while row permutations may be. In softmax classification heads and MoE routers, there is also a shared-shift invariance: adding the same scalar to all logits leaves the resulting probability vector unchanged. These structures motivate one-sided spectral updates, row-normalized updates, centered updates, and hybrid row-norm/spectral updates whose equivariance matches the actual symmetry group of the matrix block.

Our motivating application is optimizer design for modern neural network training, where different matrix-valued parameter classes carry different symmetry structures. Linear and attention projections are naturally associated with bi-orthogonal geometry; embeddings and output heads combine discrete vocabulary or label indexing with feature-space geometry; gated MLP projections exhibit intermediate-coordinate permutation symmetries; and MoE routers combine expert-permutation symmetry with shared-logit-shift invariance. These examples illustrate the broader optimization principle: when a matrix variable carries a meaningful symmetry or quotient structure, its update class should be chosen to respect that structure rather than assumed to follow a single universal entrywise rule.

We study first-order matrix optimization methods of the form

$$(\forall k \in \mathbb{N}) \qquad W_{k+1} = W_k - \gamma_k \mathcal{U}(D_k), \tag{1.1}$$

where $W_k \in \mathbb{R}^{m\times n}$ is a matrix variable, $\gamma_k > 0$ is a stepsize, $D_k \in \mathbb{R}^{m\times n}$ is a first-order direction such as a gradient or momentum direction, and $\mathcal{U}$ is an equivariant matrix update map. Depending on the symmetry group, $\mathcal{U}$ may be a full spectral, one-sided spectral, row-normalized, centered, or hybrid update. We establish descent and convergence guarantees under smoothness, alignment, and Polyak–Łojasiewicz conditions, including variants with momentum directions and inexact spectral or polar oracles.

Relationship with companion work. A companion manuscript [37] develops the symmetry-compatible principle primarily as an optimizer-design framework for modern neural network architectures. That work emphasizes layerwise optimizer routing for embeddings, LM heads, and MoE routers, together with implementation details and large-scale pre-training experiments. The present manuscript substantially reorganizes and formalizes the material for a mathematical optimization audience: it treats matrix-valued first-order methods as the primary object of study, characterizes equivariant update classes, develops the spectral and one-sided operator viewpoint, and proves convergence guarantees through alignment and update-size conditions. Thus the two manuscripts have different scopes and audiences, while sharing the same motivating principle. This extended arXiv version containing additional implementation details, ablations, and experiments is cited where appropriate.

1.1 Related work

Matrix-gradient optimizers. The empirical success of Muon [31], especially in the `modded-nanogpt` speedrun [30], has renewed interest in matrix-gradient optimizers for deep learning. This has led to a growing literature on geometry-aware and matrix-structured methods [10, 11, 15, 19, 22, 28, 29, 35, 36, 56, 58, 63, 68, 71, 72, 76, 77]. Conceptually, Muon is closely related to stochastic spectral descent [6–8]: both can be viewed through steepest descent with respect to the spectral norm, whose steepest descent direction is the orthogonal polar factor. Recent theoretical work studies local one-step behavior and convergence guarantees for Muon-type or spectral methods under various assumptions [9, 12, 23, 32, 34, 42, 48, 61, 65]. Our emphasis is different: we derive matrix-gradient update classes from symmetry and equivariance principles, and then study their descent properties through alignment and update-size bounds.

Matrix-gradient methods are also related to second-order and preconditioning approaches, including K-FAC [18, 49], Shampoo [2, 17, 25, 62], BFGS and L-BFGS methods [21], SOAP [69], KL-Shampoo and KL-SOAP [44], and preconditioned SGD [43, 57]. These methods typically approximate curvature or preconditioning structure, whereas the present paper focuses on the equivariance properties of the update map itself. Other complementary directions include weight constraints and manifold methods [3, 5, 24, 70, 73], as well as variance reduction and low-rank gradient projection methods [45, 64, 74, 78]. We refer to [55] for a broader review of geometry-aware optimization methods in deep learning.

Matrix optimization and spectral operators. Matrix optimization has long been studied as a distinct class of optimization problems [13, 14]. A central reason is that matrices carry eigenvalue, singular-value, rank, and unitary-invariance structure that is obscured by vectorization. Classical work on convex matrix analysis and spectral functions established the foundations for optimization with unitarily invariant functions and eigenvalue or singular-value structure [4, 26, 27, 38–41]. In this literature, spectral operators, also known in the rectangular case as Löwner operators [66], are matrix-valued maps that preserve singular-vector directions and act only on singular values. This is precisely the operator-theoretic structure underlying stochastic spectral descent, Muon, Scion, and polar gradient methods. Our contribution is to connect this classical spectral-operator viewpoint with symmetry-compatible first-order optimizer design for matrix variables with full, one-sided, discrete, centered, or hybrid symmetry structures.

1.2 Contributions

The main contributions of this paper are as follows.

1. We formulate a symmetry-compatible principle for matrix-gradient methods: the update rule for each matrix block should be equivariant with respect to the natural symmetry group acting on that block.

2. We characterize bi-orthogonally equivariant update maps as spectral update maps, thereby giving a unified interpretation of stochastic spectral descent, Muon, Scion, and polar gradient methods.

3. We extend the framework to one-sided, permutation, centered, and hybrid equivariance classes, yielding right-spectral, row-normalized, centered, and hybrid row-norm/spectral update maps.

4. We prove descent and convergence guarantees under smoothness and alignment assumptions, including sublinear convergence to stationarity, linear convergence under the Polyak–Łojasiewicz condition, and perturbation guarantees for momentum and inexact-oracle variants.

5. We instantiate the framework for matrix-valued parameter blocks in neural networks, leading to architecture-aware optimizer routing for embeddings, output heads, attention and linear projections, gated MLPs, and MoE routers.

1.3 Organization

Section 2 formulates the matrix optimization problem and first-order update model. Section 3 develops group-equivariant update maps, including bi-orthogonal spectral updates and one-sided, permutation, centered, and hybrid classes. Section 4 proves the convergence guarantees. Section 5 discusses architecture-aware optimizer routing for neural-network matrix blocks. Section 6 presents illustrative pre-training experiments. Section 7 concludes.

1.4 Notation

We use $\mathbb{N}$ and $\mathbb{N}^*$ for the nonnegative and positive integers. For $m, n \in \mathbb{N}^*$, let $m \wedge n \coloneqq \min\{m, n\}$. For $d \in \mathbb{N}^*$, let $\mathbf{1}_d \in \mathbb{R}^d$ denote the all-ones vector. For a set $\mathcal{C}$, let $\mathbb{1}_{\mathcal{C}}$ denote its indicator function, namely $\mathbb{1}_{\mathcal{C}}(x) = 1$ if $x \in \mathcal{C}$ and $\mathbb{1}_{\mathcal{C}}(x) = 0$ otherwise. For $s \in \mathbb{R}^{m \wedge n}$, $\mathrm{Diag}_{m,n}(s) \in \mathbb{R}^{m \times n}$ denotes the rectangular diagonal matrix with (i, i) entry s_i for $1 \leqslant i \leqslant m \wedge n$ and all other entries zero. For a square matrix S, $\mathrm{Diag}(S)$ denotes the diagonal matrix formed from its diagonal, and $\mathrm{tr}(S)$ denotes its trace. For $x \in \mathbb{R}^d$, $\mathrm{Diag}(x)$ is the diagonal matrix with diagonal x. For matrices $A, B \in \mathbb{R}^{m \times n}$, $\langle\!\langle A, B \rangle\!\rangle_{\mathrm{F}} \coloneqq \mathrm{tr}(A^\top B)$ denotes the Frobenius inner product and $|||A|||_{\mathrm{F}} \coloneqq \sqrt{\langle\!\langle A, A \rangle\!\rangle_{\mathrm{F}}}$ the Frobenius norm. We write $\sigma(A) = (\sigma_1(A), \ldots, \sigma_{m \wedge n}(A))^\top$ for the singular values in nonincreasing order, $|||A|||_{\mathrm{S}}$ for the spectral norm, and $|||A|||_{\mathrm{nuc}}$ for the nuclear norm. Let $\mathbb{S}^d$, $\mathbb{S}^d_+$, and $\mathbb{S}^d_{++}$ denote the spaces of symmetric, symmetric positive semidefinite, and symmetric positive definite $d \times d$ matrices. Let $\mathbb{O}^d \coloneqq \{Q \in \mathbb{R}^{d \times d} : Q^\top Q = QQ^\top = I_d\}$ be the orthogonal group and $\mathbb{P}^d \coloneqq \{P \in \{0, 1\}^{d \times d} : P\mathbf{1}_d = \mathbf{1}_d,\ P^\top \mathbf{1}_d = \mathbf{1}_d\}$ the set of permutation matrices. For a Euclidean space $\mathcal{E}$, $\langle \cdot, \cdot \rangle$ denotes its inner product. For an extended-real-valued function $f \colon \mathcal{E} \to \overline{\mathbb{R}} \coloneqq \mathbb{R} \cup \{\pm\infty\}$, $\mathrm{dom}\, f \coloneqq \{x : f(x) < \infty\}$, and f is proper if $\mathrm{dom}\, f \neq \emptyset$. For a nonempty closed convex set $\mathcal{C}$, $\iota_{\mathcal{C}}$ denotes its convex indicator function, equal to 0 on $\mathcal{C}$ and $+\infty$ outside $\mathcal{C}$, and $\mathrm{proj}_{\mathcal{C}}$ denotes the Euclidean projection onto $\mathcal{C}$.

2 Problem formulation

Let $m, n \in \mathbb{N}^*$ and consider

$$\underset{W \in \mathbb{R}^{m\times n}}{\text{minimize}}\ f(W),$$

where $f\colon \mathbb{R}^{m\times n} \to \mathbb{R}$ is differentiable. We equip $\mathbb{R}^{m\times n}$ with the Frobenius inner product $\langle\!\langle W, Z\rangle\!\rangle_{\mathrm{F}}$ and norm $|||W|||_{\mathrm{F}}$. The gradient $\nabla f(W)$ is defined by $\mathrm{D}f(W)[D] = \langle\!\langle \nabla f(W), D\rangle\!\rangle_{\mathrm{F}}$ for every $D \in \mathbb{R}^{m\times n}$.

We study first-order methods of the form (1.1). In the memoryless case, $D_k = \nabla f(W_k)$ and

$$(\forall k \in \mathbb{N}) \qquad W_{k+1} = W_k - \gamma_k \mathcal{U}(\nabla f(W_k)). \tag{2.1}$$

More generally, D_k may be a momentum or filtered gradient direction. The update map $\mathcal{U}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ transforms the raw gradient or filtered direction into a geometry-aware matrix update.

The central question is how to choose $\mathcal{U}$ when W is genuinely matrix-valued. Coordinatewise methods treat W as a vector after choosing an ordering of its entries. By contrast, a matrix-gradient method should respect the row, column, spectral, and symmetry structure of W. We therefore require $\mathcal{U}$ to be compatible with the natural group action associated with the matrix variable: if a group $\mathcal{G}$ acts on $\mathbb{R}^{m\times n}$ by $D \mapsto g \cdot D$, then $\mathcal{U}$ should satisfy

$$(\forall g \in \mathcal{G})(\forall D \in \mathbb{R}^{m\times n}) \qquad \mathcal{U}(g \cdot D) = g \cdot \mathcal{U}(D).$$

Different choices of $\mathcal{G}$ yield different symmetry-compatible update classes. For linear-operator matrices, $\mathcal{G} = \mathbb{O}^m \times \mathbb{O}^n$ acts by $D \mapsto PDQ^\top$. For matrices with indexed rows, grouped coordinates, or shared-shift invariances, the relevant action may involve permutations, one-sided orthogonal transformations, centering, or combinations thereof. The next sections develop these update classes and analyze their convergence.

3 Group-equivariant matrix-gradient methods

We now formalize the symmetry-compatible viewpoint. Let $\mathcal{G}$ be a group acting on $\mathbb{R}^{m\times n}$ through $(g, D) \mapsto g \cdot D$, where $g \cdot D$ represents a change of coordinates, relabeling, or symmetry transformation of the matrix direction D.

Definition 3.1 (Group-equivariant update maps)**.** Let $\mathcal{G}$ act on $\mathbb{R}^{m\times n}$. A map $\mathcal{U}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is called $\mathcal{G}$-equivariant if

$$(\forall g \in \mathcal{G})(\forall D \in \mathbb{R}^{m\times n}) \qquad \mathcal{U}(g \cdot D) = g \cdot \mathcal{U}(D).$$

We denote by $\mathcal{U}_{\mathcal{G}}^{m\times n}$ the class of all $\mathcal{G}$-equivariant update maps from $\mathbb{R}^{m\times n}$ to itself.

A first-order method of the form (1.1) is called $\mathcal{G}$-symmetry-compatible if $\mathcal{U} \in \mathcal{U}_{\mathcal{G}}^{m\times n}$ and the direction sequence $(D_k)_{k\in\mathbb{N}}$ transforms equivariantly under the same group action. In the memoryless case, this gives (2.1). The role of equivariance is to ensure that the update does not depend on an arbitrary representation of the matrix direction.

The appropriate group depends on the matrix variable. For a matrix representing a linear operator between Euclidean spaces, the natural group is $\mathbb{O}^m \times \mathbb{O}^n$, acting by

$$(P, Q) \cdot D = PDQ^\top, \qquad P \in \mathbb{O}^m, \quad Q \in \mathbb{O}^n.$$

For matrices whose rows index discrete objects, such as categories, tokens, or experts, arbitrary row rotations are not meaningful, while row permutations may be. This leads to actions such as $\mathbb{P}^m$ or $\mathbb{P}^m \times \mathbb{O}^n$. In other cases, the relevant symmetry includes centering or shared-shift invariance. The rest of this section studies the update classes induced by these choices of $\mathcal{G}$.

3.1 Bi-orthogonal equivariance and spectral operators

We first consider matrix variables that represent linear operators. If the input and output bases are changed by $Q \in \mathbb{O}^n$ and $P \in \mathbb{O}^m$, then an update direction $D \in \mathbb{R}^{m\times n}$ is represented as $PDQ^\top$. A coordinate-free update map should therefore commute with this left-right orthogonal action.

Definition 3.2 (Bi-orthogonally equivariant update maps)**.** A map $\mathfrak{U}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is called bi-orthogonally equivariant if

$$(\forall P \in \mathbb{O}^m)(\forall Q \in \mathbb{O}^n)(\forall D \in \mathbb{R}^{m\times n}) \qquad \mathfrak{U}(PDQ^\top) = P\mathfrak{U}(D)Q^\top.$$

We denote this class by $\mathcal{U}_{\mathbb{O}}^{m\times n} \coloneqq \mathcal{U}_{\mathbb{O}^m\times\mathbb{O}^n}^{m\times n}$.

Bi-orthogonal equivariance means that the update rule is independent of the orthonormal bases used to represent the underlying linear operator. The corresponding operator class is the class of spectral update maps.

Definition 3.3 (Spectral update maps)**.** Let $r \coloneqq m \wedge n$. A map $\mathfrak{U}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is called a spectral update map if there exists a map $\psi\colon \mathbb{R}^r_+ \to \mathbb{R}^r$ such that, for every singular value decomposition $D = U\,\mathrm{Diag}_{m,n}(\sigma(D))V^\top$,

$$\mathfrak{U}(D) = U\,\mathrm{Diag}_{m,n}(\psi(\sigma(D)))V^\top, \tag{3.1}$$

where ψ is permutation-equivariant,

$$(\forall P \in \mathbb{P}^r)(\forall s \in \mathbb{R}^r_+) \qquad \psi(Ps) = P\psi(s),$$

and satisfies the zero-support condition

$$(\forall s \in \mathbb{R}^r_+)(\forall i \in \{1,\dots,r\}) \qquad s_i = 0 \quad \Rightarrow \quad \psi_i(s) = 0. \tag{3.2}$$

Permutation equivariance and the zero-support condition ensure that (3.1) is independent of the chosen singular value decomposition. The former handles relabeling and repeated singular-value subspaces, while the latter handles the nonuniqueness of singular vectors associated with zero singular values. Thus spectral update maps are matrix-valued transformations of singular values, rather than coordinatewise transformations of matrix entries.

Remark 3.1 (Relation with absolute symmetry). For scalar spectral functions, the singular-value function is usually required to be absolutely symmetric. For vector-valued spectral update maps, the analogous condition is equivariance rather than invariance. Equivalently, one may define ψ on $\mathbb{R}^r$ and require signed-permutation equivariance, $\psi(Ss) = S\psi(s)$ for every signed permutation matrix S. Since singular values lie in $\mathbb{R}^r_+$, this reduces to permutation equivariance on $\mathbb{R}^r_+$ together with (3.2).

Theorem 3.1 (Characterization of bi-orthogonally equivariant update maps). *A map $\mathfrak{U}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is bi-orthogonally equivariant if and only if it is a spectral update map in the sense of Definition 3.3. Moreover, $\mathfrak{U}$ is continuous if and only if the associated singular-value map ψ can be chosen continuous on $\mathbb{R}^r_+$.*

Proof. If $\mathfrak{U}$ has the spectral form (3.1), then for any $P \in \mathbb{O}^m$ and $Q \in \mathbb{O}^n$, the matrix $PDQ^\top$ has singular value decomposition $(PU)\,\mathrm{Diag}_{m,n}(\sigma(D))(QV)^\top$. Hence

$$\mathfrak{U}(PDQ^\top) = (PU)\,\mathrm{Diag}_{m,n}(\psi(\sigma(D)))(QV)^\top = P\mathfrak{U}(D)Q^\top.$$

Thus $\mathfrak{U}$ is bi-orthogonally equivariant.

Conversely, suppose $\mathfrak{U}$ is bi-orthogonally equivariant. Let $r \coloneqq m \wedge n$ and, for $s \in \mathbb{R}^r_+$, write $D_s \coloneqq \mathrm{Diag}_{m,n}(s)$. We first show that $\mathfrak{U}(D_s)$ is rectangular diagonal. Let $S = \mathrm{Diag}(\epsilon_1, \dots, \epsilon_m) \in \mathbb{O}^m$ and $T = \mathrm{Diag}(\delta_1, \dots, \delta_n) \in \mathbb{O}^n$, where $\epsilon_i, \delta_j \in \{\pm 1\}$ and $\epsilon_i = \delta_i$ for $1 \leqslant i \leqslant r$. Then $SD_sT^\top = D_s$, so equivariance gives $\mathfrak{U}(D_s) = S\mathfrak{U}(D_s)T^\top$. Entrywise,

$$\big(\mathfrak{U}(D_s)\big)_{ij} = \epsilon_i\delta_j\big(\mathfrak{U}(D_s)\big)_{ij}.$$

If $i \neq j$, or if $i > r$ or $j > r$, the signs can be chosen so that $\epsilon_i\delta_j = -1$ while preserving $\epsilon_\ell = \delta_\ell$ for $1 \leqslant \ell \leqslant r$. Hence all such entries vanish. Therefore there exists $\psi(s) \in \mathbb{R}^r$ such that

$$\mathfrak{U}(D_s) = \mathrm{Diag}_{m,n}(\psi(s)). \tag{3.3}$$

Let $P \in \mathbb{P}^r$ and extend it to compatible permutation matrices $\overline{P}_m \in \mathbb{P}^m$ and $\overline{P}_n \in \mathbb{P}^n$ acting as P on the first r coordinates and as the identity elsewhere. Since $\mathrm{Diag}_{m,n}(Ps) = \overline{P}_m D_s \overline{P}_n^\top$, equivariance and (3.3) imply

$$\mathrm{Diag}_{m,n}(\psi(Ps)) = \overline{P}_m\,\mathrm{Diag}_{m,n}(\psi(s))\overline{P}_n^\top = \mathrm{Diag}_{m,n}(P\psi(s)).$$

Thus $\psi(Ps) = P\psi(s)$.

The zero-support condition is also forced by equivariance. If $s_i = 0$, let $S \in \mathbb{O}^m$ be the diagonal sign matrix with $S_{ii} = -1$ and all other diagonal entries equal to 1, and let $T = I_n$. Then $SD_sT^\top = D_s$, so

$$\mathrm{Diag}_{m,n}(\psi(s)) = \mathfrak{U}(D_s) = S\mathfrak{U}(D_s)T^\top = S\,\mathrm{Diag}_{m,n}(\psi(s)).$$

Comparing the (i,i) entry gives $\psi_i(s) = 0$.

Finally, for any $D = U\,\mathrm{Diag}_{m,n}(\sigma(D))V^\top$, equivariance and (3.3) yield

$$\mathfrak{U}(D) = U\mathfrak{U}(\mathrm{Diag}_{m,n}(\sigma(D)))V^\top = U\,\mathrm{Diag}_{m,n}(\psi(\sigma(D)))V^\top.$$

The formula is independent of the chosen singular value decomposition because it is derived from $\mathfrak{U}$ itself; equivalently, stabilizer transformations of $\mathrm{Diag}_{m,n}(\sigma(D))$ also stabilize $\mathrm{Diag}_{m,n}(\psi(\sigma(D)))$. The continuity statement follows by restriction to rectangular diagonal matrices and continuity of the spectral reconstruction map under the compatibility conditions above. □

This characterization explains why spectral matrix-gradient methods arise from the coordinate-free consistency requirement. The identity map $\psi(s) = s$ gives ordinary gradient descent, while the polar choice $\psi(s) = \mathbb{1}_{\{s>0\}}$ gives the orthogonal polar factor. Stochastic spectral descent, Muon, Scion, and polar gradient methods all fit this framework, possibly after applying the spectral operator to a momentum direction rather than the raw gradient.

It also clarifies the equivariance limitation of coordinatewise adaptive updates for matrix variables with bi-orthogonal symmetry. Entrywise nonlinearities are generally equivariant under coordinate relabelings and sign changes, but not under arbitrary left and right orthogonal changes of basis. Thus, for variables whose natural symmetry is bi-orthogonal, such updates do not in general define spectral operators in the sense of Definition 3.3. This does not preclude their use for parameters with fixed coordinate semantics, but it distinguishes them from the coordinate-free update maps studied here.

3.2 One-sided, permutation, and centered equivariance classes

We next consider matrix variables whose natural symmetry group is smaller than the full left-right orthogonal group. This occurs when one axis has a distinguished meaning, such as indexing categories, tokens, experts, or other discrete objects. In such cases, arbitrary rotations along that axis are not meaningful, while permutations are. The resulting update classes are not fully spectral, but remain equivariant under the transformations that preserve the meaning of the matrix variable.

3.2.1 Left-permutation/right-orthogonal maps

Let $D \in \mathbb{R}^{v\times d}$ have rows indexing discrete objects and columns representing a Euclidean feature space. The natural group is $\mathbb{P}^v \times \mathbb{O}^d$, acting by

$$(P, R) \cdot D = PDR^\top, \qquad P \in \mathbb{P}^v, \quad R \in \mathbb{O}^d.$$

The corresponding equivariance condition is

$$(\forall P \in \mathbb{P}^v)(\forall R \in \mathbb{O}^d)(\forall D \in \mathbb{R}^{v\times d}) \qquad \mathcal{U}(PDR^\top) = P\mathcal{U}(D)R^\top. \tag{3.4}$$

We call such maps left-permutation/right-orthogonal equivariant, or LPRO-equivariant, and write $\mathcal{U}^{v\times d}_{\mathsf{LPRO}} \coloneqq \mathcal{U}^{v\times d}_{\mathbb{P}^v\times\mathbb{O}^d}$.

3.2.2 Right-spectral maps

A basic LPRO-equivariant construction uses the right Gram matrix $D^\top D$. We say that $\mathcal{U}_{\mathsf{R}}\colon \mathbb{R}^{v\times d} \to \mathbb{R}^{v\times d}$ is right-spectral if

$$\mathcal{U}_{\mathsf{R}}(D) = D\Phi(D^\top D), \tag{3.5}$$

where $\Phi\colon \mathbb{S}^d_+ \to \mathbb{R}^{d\times d}$ is orthogonally equivariant:

$$(\forall R \in \mathbb{O}^d)(\forall S \in \mathbb{S}^d_+) \qquad \Phi(RSR^\top) = R\Phi(S)R^\top. \tag{3.6}$$

Since left permutations leave $D^\top D$ invariant and right orthogonal transformations conjugate it, right-spectral maps satisfy (3.4).

Proposition 3.1.1 (Right-spectral maps are LPRO-equivariant). *Let $\mathcal{U}_{\mathsf{R}}$ be defined by (3.5), where Φ satisfies (3.6). Then $\mathcal{U}_{\mathsf{R}} \in \mathcal{U}^{v\times d}_{\mathsf{LPRO}}$.*

The choice $\Phi(X) = X^{-1/2}$ gives the right polar factor $D(D^\top D)^{-1/2}$ on the support of $D^\top D$. Damped or approximate variants replace $X^{-1/2}$ by $(X + \varepsilon I_d)^{-1/2}$ for a small $\varepsilon > 0$ or by a numerical inverse-square-root oracle.

3.2.3 Row-normalized maps

Since left permutations relabel rows and right orthogonal transformations preserve row norms, row-wise normalization is also LPRO-equivariant. For $D \in \mathbb{R}^{v\times d}$, let $r_i(D) \coloneqq \|D_{i,:}\|_2$. Given $\eta\colon \mathbb{R}_+ \to \mathbb{R}$, define

$$\mathcal{U}_{\mathsf{row}}(D) = \mathrm{Diag}(\eta(r_1(D)), \dots, \eta(r_v(D)))D. \tag{3.7}$$

A typical choice is $\eta(t) = (t + \varepsilon)^{-1}$ for a small $\varepsilon > 0$.

Proposition 3.1.2 (Row-normalized maps are LPRO-equivariant). *The map $\mathcal{U}_{\mathsf{row}}$ in (3.7) belongs to $\mathcal{U}^{v\times d}_{\mathsf{LPRO}}$.*

Row-normalized maps are generally not bi-orthogonally equivariant, because arbitrary left orthogonal transformations do not preserve individual row norms. They are nevertheless equivariant under the smaller group $\mathbb{P}^v \times \mathbb{O}^d$.

3.2.4 Hybrid maps

Right-spectral maps use global feature-space geometry, while row-normalized maps use row-wise scale information. These two constructions can be combined by first applying a permutation-equivariant row scaling and then applying a right-spectral transformation to the scaled direction. Let $\mathsf{A}(D) \in \mathbb{R}^{v\times v}$ be a diagonal row-scaling matrix satisfying

$$\mathsf{A}(PDR^\top) = P\mathsf{A}(D)P^\top \tag{3.8}$$

for all $P \in \mathbb{P}^v$, $R \in \mathbb{O}^d$, and $D \in \mathbb{R}^{v\times d}$. Define $\mathsf{Y}(D) \coloneqq \mathsf{A}(D)D$. A hybrid row-norm/right-spectral map is then

$$\mathcal{U}_{\mathsf{hyb}}(D) = \mathsf{Y}(D)\Phi(\mathsf{Y}(D)^\top\mathsf{Y}(D)), \tag{3.9}$$

where $\Phi\colon \mathbb{S}^d_+ \to \mathbb{R}^{d\times d}$ is orthogonally equivariant in the sense of (3.6). For example, one may take $\mathsf{A}(D)$ as $\mathcal{U}_{\mathsf{row}}(D)$ defined in (3.7).

Proposition 3.1.3 (Hybrid maps are LPRO-equivariant). *Let $\mathcal{U}_{\mathsf{hyb}}$ be defined by (3.9). Suppose that A satisfies (3.8) and that Φ is orthogonally equivariant. Then $\mathcal{U}_{\mathsf{hyb}} \in \mathcal{U}^{v\times d}_{\mathsf{LPRO}}$.*

Proof. Let $P \in \mathbb{P}^v$, $R \in \mathbb{O}^d$, and $D \in \mathbb{R}^{v\times d}$. By (3.8),

$$\mathsf{Y}(PDR^\top) = \mathsf{A}(PDR^\top)PDR^\top = P\mathsf{A}(D)P^\top PDR^\top = P\mathsf{Y}(D)R^\top.$$

Therefore, $\mathsf{Y}(PDR^\top)^\top\mathsf{Y}(PDR^\top) = R\mathsf{Y}(D)^\top\mathsf{Y}(D)R^\top$. Using the orthogonal equivariance of Φ, we obtain

$$\begin{aligned}
\mathcal{U}_{\mathsf{hyb}}(PDR^\top) &= \mathsf{Y}(PDR^\top)\Phi\Big(\mathsf{Y}(PDR^\top)^\top\mathsf{Y}(PDR^\top)\Big)\\
&= P\mathsf{Y}(D)R^\top\Phi\Big(R\mathsf{Y}(D)^\top\mathsf{Y}(D)R^\top\Big)\\
&= P\mathsf{Y}(D)R^\top R\Phi(\mathsf{Y}(D)^\top\mathsf{Y}(D))R^\top\\
&= P\mathcal{U}_{\mathsf{hyb}}(D)R^\top.
\end{aligned}$$

Thus $\mathcal{U}_{\mathsf{hyb}}$ is LPRO-equivariant. □

3.2.5 Centered maps

Some matrix variables possess an additional shared-shift invariance. In a softmax classification head or MoE router, adding the same scalar to all logits leaves the induced probability vector unchanged. Hence only directions orthogonal to the all-ones direction are identifiable in the row dimension.

Let $\Pi^\perp_e \coloneqq I_e - \frac{1}{e}\mathbf{1}_e\mathbf{1}_e^\top$ be the projector onto the subspace orthogonal to $\mathbf{1}_e$. A centered update map has output in the centered row subspace: $\mathbf{1}_e^\top\mathcal{U}(D) = 0$ for all $D \in \mathbb{R}^{e\times d}$. A general construction is

$$\mathcal{U}_{\mathsf{ctr}}(D) = \Pi^\perp_e\mathcal{V}(\Pi^\perp_e D),$$

where $\mathcal{V}\colon \mathbb{R}^{e\times d} \to \mathbb{R}^{e\times d}$ is a base update map. The first projection removes the shared-shift component of the input direction, while the final projection ensures that the update remains tangent to the centered quotient representative.

Since $\Pi^\perp_e P = P\Pi^\perp_e$ for every $P \in \mathbb{P}^e$, centering is compatible with row permutations. For instance, a centered row-normalized update is

$$\mathcal{U}_{\mathsf{crow}}(D) = \Pi^\perp_e\,\mathrm{Diag}(\eta(r_1(\Pi^\perp_e D)),\dots,\eta(r_e(\Pi^\perp_e D)))\Pi^\perp_e D,$$

and a centered left-spectral update is $\mathcal{U}_{\mathsf{cL}}(D) = \Pi^\perp_e\Phi((\Pi^\perp_e D)(\Pi^\perp_e D)^\top)\Pi^\perp_e D$. These centered maps encode both expert or class relabeling symmetry and shared-logit-shift invariance.

4 Convergence analysis

We now give a convergence framework for symmetry-compatible matrix-gradient methods. The key point is that equivariance determines the admissible form of the update map, while descent depends only on two scalar quantities: the alignment between the gradient and the update direction, and the size of the update. This separation allows the same analysis to cover spectral, one-sided spectral, row-normalized, centered, hybrid, momentum, and inexact-oracle variants.

4.1 A general descent template

We consider the deterministic iteration

$$(\forall k \in \mathbb{N}) \qquad W_{k+1} = W_k - \gamma_k \mathcal{U}(D_k), \tag{4.1}$$

where $W_k \in \mathbb{R}^{m\times n}$, $\gamma_k > 0$, $D_k \in \mathbb{R}^{m\times n}$ is a gradient or filtered-gradient direction, and $\mathcal{U}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is a symmetry-compatible update map. The memoryless case corresponds to $D_k = \nabla f(W_k)$.

Assumption 4.1 (L-smoothness)**.** The function $f\colon \mathbb{R}^{m\times n} \to \mathbb{R}$ is differentiable and L-smooth with respect to the Frobenius norm:

$$(\forall W, Z \in \mathbb{R}^{m\times n}) \qquad f(Z) \leqslant f(W) + \langle\!\langle \nabla f(W), Z - W \rangle\!\rangle_{\mathrm{F}} + \frac{L}{2}|||Z - W|||_{\mathrm{F}}^2.$$

Assumption 4.2 (Lower boundedness)**.** The function f is bounded below on $\mathbb{R}^{m\times n}$, and $f^\star := \inf_{W\in\mathbb{R}^{m\times n}} f(W) > -\infty$.

For each iteration, we define $a_k := \langle\!\langle \nabla f(W_k), \mathcal{U}(D_k) \rangle\!\rangle_{\mathrm{F}}$ and $b_k := |||\mathcal{U}(D_k)|||_{\mathrm{F}}^2$. The scalar a_k measures descent alignment, while b_k controls the quadratic smoothness penalty.

Lemma 4.0.1 (One-step descent). *Suppose that Assumption 4.1 holds. Then (4.1) satisfies*

$$(\forall k \in \mathbb{N}) \qquad f(W_{k+1}) \leqslant f(W_k) - \gamma_k a_k + \frac{L\gamma_k^2}{2} b_k.$$

Consequently, if $a_k > 0$, $b_k > 0$, and $0 < \gamma_k < 2a_k/(Lb_k)$, then $f(W_{k+1}) < f(W_k)$.

Proof. Apply Assumption 4.1 with $Z = W_{k+1} = W_k - \gamma_k \mathcal{U}(D_k)$ to obtain

$$f(W_{k+1}) \leqslant f(W_k) - \gamma_k \langle\!\langle \nabla f(W_k), \mathcal{U}(D_k) \rangle\!\rangle_{\mathrm{F}} + \frac{L\gamma_k^2}{2}|||\mathcal{U}(D_k)|||_{\mathrm{F}}^2,$$

which is the claimed bound. The strict descent statement follows immediately. □

The memoryless convergence results follow from the following uniform alignment and update-size condition.

Assumption 4.3 (Gradient alignment and update-size control)**.** There exist constants $\alpha > 0$ and $\beta > 0$ such that, for every $W \in \mathbb{R}^{m\times n}$,

$$\langle\!\langle \nabla f(W), \mathcal{U}(\nabla f(W)) \rangle\!\rangle_{\mathrm{F}} \geqslant \alpha |||\nabla f(W)|||_{\mathrm{F}}^2, \qquad |||\mathcal{U}(\nabla f(W))|||_{\mathrm{F}}^2 \leqslant \beta |||\nabla f(W)|||_{\mathrm{F}}^2.$$

Assumption 4.4 (Polyak–Łojasiewicz condition)**.** The function f satisfies the μ-Polyak–Łojasiewicz condition for some $\mu > 0$:

$$(\forall W \in \mathbb{R}^{m\times n}) \qquad \frac{1}{2}|||\nabla f(W)|||_{\mathrm{F}}^2 \geqslant \mu(f(W) - f^\star).$$

Theorem 4.1 (Convergence under alignment and update-size control). *Suppose that Assumptions 4.1 to 4.3 hold. Consider the memoryless method*

$$(\forall k \in \mathbb{N}) \qquad W_{k+1} = W_k - \gamma \mathfrak{U}(\nabla f(W_k)),$$

with constant stepsize $0 < \gamma < 2\alpha/(L\beta)$. *Then, for every* $K \in \mathbb{N}^*$,

$$\min_{0 \leqslant k \leqslant K-1} |||\nabla f(W_k)|||_{\mathrm{F}}^2 \leqslant \frac{f(W_0) - f^\star}{\gamma(\alpha - L\gamma\beta/2)K}. \tag{4.2}$$

If, in addition, Assumption 4.4 holds, then

$$(\forall k \in \mathbb{N}) \qquad f(W_k) - f^\star \leqslant \left(1 - 2\mu\gamma\left(\alpha - \frac{L\gamma\beta}{2}\right)\right)^k (f(W_0) - f^\star). \tag{4.3}$$

In particular, the choice $\gamma = \alpha/(L\beta)$ *gives the stationarity bound*

$$\min_{0 \leqslant k \leqslant K-1} |||\nabla f(W_k)|||_{\mathrm{F}}^2 \leqslant \frac{2L\beta(f(W_0) - f^\star)}{\alpha^2 K}$$

and the PŁ rate

$$(\forall k \in \mathbb{N}) \qquad f(W_k) - f^\star \leqslant \left(1 - \frac{\mu\alpha^2}{L\beta}\right)^k (f(W_0) - f^\star).$$

Proof. By Lemma 4.0.1 and Assumption 4.3, the memoryless iteration satisfies

$$f(W_{k+1}) \leqslant f(W_k) - \gamma\left(\alpha - \frac{L\gamma\beta}{2}\right) |||\nabla f(W_k)|||_{\mathrm{F}}^2.$$

Since $0 < \gamma < 2\alpha/(L\beta)$, the coefficient $\gamma(\alpha - L\gamma\beta/2)$ is positive. Rearranging and summing from $k = 0$ to $K - 1$ gives

$$\gamma\left(\alpha - \frac{L\gamma\beta}{2}\right) \sum_{k=0}^{K-1} |||\nabla f(W_k)|||_{\mathrm{F}}^2 \leqslant f(W_0) - f(W_K) \leqslant f(W_0) - f^\star.$$

Therefore, (4.2) follows. If the μ-PŁ condition holds, then

$$f(W_{k+1}) - f^\star \leqslant \left(1 - 2\mu\gamma\left(\alpha - \frac{L\gamma\beta}{2}\right)\right)(f(W_k) - f^\star),$$

and iterating the recursion yields the claimed linear rate in (4.3). □

Remark 4.1 (On stochastic gradients). In large-scale learning, D_k is usually a stochastic gradient or a filtered stochastic direction. Since the update maps considered here are nonlinear, unbiasedness of a stochastic gradient estimator does not imply unbiasedness of $\mathfrak{U}(D_k)$. A stochastic theory therefore requires assumptions directly on expected alignment and update-size quantities, and is left for future work.

4.2 Verification for symmetry-compatible update classes

We now record how the update classes in Section 3 fit into Assumption 4.3. The corresponding alignment and update-size calculations are collected in Section A.2.

Spectral update maps. Let $r = m \wedge n$ and let $\mathcal{U}$ be a spectral update map such that $\mathcal{U}(G) = U \operatorname{Diag}_{m,n}(\psi(\sigma(G)))V^\top$ and $G = U \operatorname{Diag}_{m,n}(\sigma(G))V^\top$. Then we have

$$\langle\!\langle G, \mathcal{U}(G) \rangle\!\rangle_{\mathrm{F}} = \langle \sigma(G), \psi(\sigma(G)) \rangle, \qquad |\!|\!| \mathcal{U}(G) |\!|\!|_{\mathrm{F}}^2 = \|\psi(\sigma(G))\|_2^2.$$

Consequently, if there exist $\alpha, \beta > 0$ such that

$$(\forall s \in \mathbb{R}^r_+) \qquad \langle s, \psi(s) \rangle \geqslant \alpha \|s\|_2^2, \qquad \|\psi(s)\|_2^2 \leqslant \beta \|s\|_2^2, \tag{4.4}$$

then Assumption 4.3 holds and Theorem 4.1 applies.

The identity map has $\alpha = \beta = 1$. The polar map $\psi_i(s) = \mathbb{1}_{\{s_i > 0\}}$ is normalized and does not satisfy uniform update-size control near the origin. Instead, we have $\langle\!\langle G, \operatorname{polar}(G) \rangle\!\rangle_{\mathrm{F}} = |\!|\!| G |\!|\!|_{\mathrm{nuc}}$ and $|\!|\!| \operatorname{polar}(G) |\!|\!|_{\mathrm{F}}^2 = \operatorname{rank}(G)$, so it is naturally analyzed through Lemma 4.0.1 with adaptive or scale-aware stepsizes. A scale-aware variant is $\mathcal{U}_{\mathsf{nsp}}(G) = \frac{|\!|\!| G |\!|\!|_{\mathrm{nuc}}}{r} \operatorname{polar}(G)$, which satisfies Assumption 4.3 with $\alpha = 1/r$ and $\beta = 1$.

One-sided spectral update maps. For a right-spectral map $\mathcal{U}_{\mathsf{R}}(G) = G\Phi(G^\top G)$, write $G^\top G = V \operatorname{Diag}(t) V^\top$, where $t = (t_1, \ldots, t_d) \in \mathbb{R}^d_+$ and $t_i = \lambda_i(G^\top G) = \sigma_i(G)^2$, with zero padding if necessary. If $\Phi(G^\top G) = V \operatorname{Diag}(\phi(t)) V^\top$, then

$$\langle\!\langle G, \mathcal{U}_{\mathsf{R}}(G) \rangle\!\rangle_{\mathrm{F}} = \sum_{i=1}^d t_i \phi_i(t), \qquad |\!|\!| \mathcal{U}_{\mathsf{R}}(G) |\!|\!|_{\mathrm{F}}^2 = \sum_{i=1}^d t_i \phi_i(t)^2.$$

Thus Assumption 4.3 holds if there exist $\alpha, \beta > 0$ such that, for every $t \in \mathbb{R}^d_+$, $\sum_{i=1}^d t_i \phi_i(t) \geqslant \alpha \sum_{i=1}^d t_i$ and $\sum_{i=1}^d t_i \phi_i(t)^2 \leqslant \beta \sum_{i=1}^d t_i$. The left-spectral case follows by transposition.

Row-normalized update maps. For $\mathcal{U}_{\mathsf{row}}(G) = \operatorname{Diag}(\eta(r_1(G)), \ldots, \eta(r_v(G)))G$, where $r_i(G) = \|G_{i,:}\|_2$, we have

$$\langle\!\langle G, \mathcal{U}_{\mathsf{row}}(G) \rangle\!\rangle_{\mathrm{F}} = \sum_{i=1}^v \eta(r_i(G)) r_i(G)^2, \qquad |\!|\!| \mathcal{U}_{\mathsf{row}}(G) |\!|\!|_{\mathrm{F}}^2 = \sum_{i=1}^v \eta(r_i(G))^2 r_i(G)^2.$$

Therefore Assumption 4.3 holds if there exist $\alpha, \beta > 0$ such that, for every $t \in \mathbb{R}_+$, $\eta(t) t^2 \geqslant \alpha t^2$ and $\eta(t)^2 t^2 \leqslant \beta t^2$. Exact row normalization, like the polar update, is normalized rather than scale-compatible. If $\mathcal{U}_{\mathsf{rownorm}}$ denotes exact row normalization, then $\langle\!\langle G, \mathcal{U}_{\mathsf{rownorm}}(G) \rangle\!\rangle_{\mathrm{F}} = \sum_{i=1}^v r_i(G)$ and $|\!|\!| \mathcal{U}_{\mathsf{rownorm}}(G) |\!|\!|_{\mathrm{F}}^2 = \sharp\{i : r_i(G) > 0\}$, so descent follows from Lemma 4.0.1 with an adaptive or scale-aware stepsize.

Centered update maps. Let $\Pi_e^\perp = I_e - \frac{1}{e} \mathbf{1}_e \mathbf{1}_e^\top$ and $\mathcal{U}_{\mathsf{ctr}}(D) = \Pi_e^\perp \mathcal{V}(\Pi_e^\perp D)$. Since $\mathcal{U}_{\mathsf{ctr}}(D)$ lies in the centered row subspace, we have

$$\langle\!\langle \nabla f(W), \mathcal{U}_{\mathsf{ctr}}(\nabla f(W)) \rangle\!\rangle_{\mathrm{F}} = \big\langle\!\big\langle \Pi_e^\perp \nabla f(W), \mathcal{U}_{\mathsf{ctr}}(\nabla f(W)) \big\rangle\!\big\rangle_{\mathrm{F}}.$$

Thus the preceding convergence theorem applies in the quotient geometry after replacing $\nabla f(W)$ by $\Pi_e^\perp \nabla f(W)$. In particular, if there exist $\alpha, \beta > 0$ such that

$$\big\langle\!\big\langle \Pi_e^\perp \nabla f(W), \mathcal{U}_{\mathsf{ctr}}(\nabla f(W)) \big\rangle\!\big\rangle_{\mathrm{F}} \geqslant \alpha |\!|\!| \Pi_e^\perp \nabla f(W) |\!|\!|_{\mathrm{F}}^2,$$
$$\text{and} \quad |\!|\!| \mathcal{U}_{\mathsf{ctr}}(\nabla f(W)) |\!|\!|_{\mathrm{F}}^2 \leqslant \beta |\!|\!| \Pi_e^\perp \nabla f(W) |\!|\!|_{\mathrm{F}}^2,$$

then

$$f(W_{k+1}) \leqslant f(W_k) - \gamma\Big(\alpha - \frac{L\gamma\beta}{2}\Big)\|\Pi_e^{\perp}\nabla f(W_k)\|_{\mathrm{F}}^2.$$

Consequently, the method converges to stationarity in the centered quotient geometry. If f is invariant under shared row shifts, then $\Pi_e^{\perp}\nabla f(W) = \nabla f(W)$ and the result reduces to the ordinary gradient-alignment framework.

Hybrid update maps. For the hybrid row-norm/right-spectral map, define the scaled direction $\mathsf{Y}(D) \coloneqq \mathsf{A}(D)D$ and set $\mathcal{U}_{\mathsf{hyb}}(D) \coloneqq \mathsf{Y}(D)\Phi(\mathsf{Y}(D)^{\top}\mathsf{Y}(D))$. A clean sufficient condition is that the row-scaling and spectral factors are positive semidefinite and uniformly controlled: for some $0 < a_{\min} \leqslant a_{\max} < \infty$ and $0 < \rho_{\min} \leqslant \rho_{\max} < \infty$,

$$a_{\min} I_v \preccurlyeq \mathsf{A}(D) \preccurlyeq a_{\max} I_v, \qquad \rho_{\min} I_d \preccurlyeq \Phi(\mathsf{Y}(D)^{\top}\mathsf{Y}(D)) \preccurlyeq \rho_{\max} I_d.$$

Then Assumption 4.3 holds with $\alpha = a_{\min}\rho_{\min}$ and $\beta = a_{\max}^2\rho_{\max}^2$, so Theorem 4.1 applies. The centered hybrid case is obtained by replacing D by $\Pi_e^{\perp}D$ and projecting the output by $\Pi_e^{\perp}$.

4.3 Momentum and inexact-oracle variants

The same descent template also covers momentum directions and inexact matrix oracles. Let $(M_k)_{k\in\mathbb{N}}$ be a filtered direction, for example, for any $k \in \mathbb{N}$, $M_k = \beta_1 M_{k-1} + (1-\beta_1)\nabla f(W_k)$ with $\beta_1 \in [0,1)$, and consider the iteration

$$(\forall k \in \mathbb{N}) \qquad W_{k+1} = W_k - \gamma\mathcal{U}(M_k).$$

Equivariance is preserved whenever the initial buffer and gradients transform equivariantly, since the momentum recursion is linear. For convergence, it suffices to assume alignment with the current gradient:

$$\langle\!\langle \nabla f(W_k), \mathcal{U}(M_k)\rangle\!\rangle_{\mathrm{F}} \geqslant \alpha_m\|\nabla f(W_k)\|_{\mathrm{F}}^2, \qquad \|\mathcal{U}(M_k)\|_{\mathrm{F}}^2 \leqslant \beta_m\|\nabla f(W_k)\|_{\mathrm{F}}^2. \tag{4.5}$$

Then Theorem 4.1 applies with α_m, β_m in place of α, β.

For inexact oracles, write $\widehat{\mathcal{U}}(D_k) = \mathcal{U}(D_k) + E_k$. If the ideal update satisfies $\langle\!\langle \nabla f(W_k), \mathcal{U}(D_k)\rangle\!\rangle_{\mathrm{F}} \geqslant \alpha\|\nabla f(W_k)\|_{\mathrm{F}}^2$ and $\|\mathcal{U}(D_k)\|_{\mathrm{F}}^2 \leqslant \beta\|\nabla f(W_k)\|_{\mathrm{F}}^2$, and the oracle error satisfies $\|E_k\|_{\mathrm{F}} \leqslant \delta\|\nabla f(W_k)\|_{\mathrm{F}}$ with $0 \leqslant \delta < \alpha$, then the inexact method $W_{k+1} = W_k - \gamma\widehat{\mathcal{U}}(D_k)$ satisfies

$$f(W_{k+1}) \leqslant f(W_k) - \gamma\left(\alpha - \delta - \frac{L\gamma(\sqrt{\beta}+\delta)^2}{2}\right)\|\nabla f(W_k)\|_{\mathrm{F}}^2.$$

Therefore the sublinear and PŁ convergence bounds hold whenever $0 < \gamma < 2(\alpha-\delta)/(L(\sqrt{\beta}+\delta)^2)$. For example, if $\mathcal{U}_{\mathsf{RPolar}}(D) = D(D^{\top}D)^{-1/2}$ and $\widehat{\mathcal{U}}(D) = D\mathsf{Q}(D)$, then

$$E = D\Big(\mathsf{Q}(D) - (D^{\top}D)^{-1/2}\Big), \qquad \|E\|_{\mathrm{F}} \leqslant \|D\|_{\mathrm{F}}\|\mathsf{Q}(D) - (D^{\top}D)^{-1/2}\|_{\mathrm{S}}.$$

Thus a spectral-norm guarantee on the inverse-square-root approximation yields a sufficient Frobenius-norm oracle-error bound.

5 Applications to optimizer design for neural network training

We now explain how the preceding equivariance framework informs optimizer design for neural network training. The purpose of this section is not to give a complete taxonomy of neural network optimizers, but to illustrate a general design principle: each matrix-valued parameter block should be optimized by an update map that commutes with the transformations preserving the functional meaning of that block. Different blocks in the same architecture may therefore require different symmetry-compatible update classes.

5.1 Blockwise symmetry and optimizer routing

Let $\mathcal{B}$ denote a collection of matrix-valued parameter blocks. A symmetry-compatible optimizer assigns to each block $b \in \mathcal{B}$ a symmetry group $\mathcal{G}_b$ and an update map $\mathcal{U}_b \in \mathcal{U}_{\mathcal{G}_b}$:

$$W_{b,k+1} = W_{b,k} - \gamma_{b,k}\mathcal{U}_b(D_{b,k}),$$

where $D_{b,k}$ is a gradient, momentum, or filtered-gradient direction. The group $\mathcal{G}_b$ should encode the transformations that preserve the meaning of the block. The update map may then be chosen from the corresponding equivariant class, such as a spectral, one-sided spectral, row-normalized, centered, or hybrid update.

A representative routing rule is summarized in Table 1. The table is intended as a design guide rather than a rigid prescription. The central requirement is structural: if a parameter block is transformed by a symmetry of the architecture, then the update should transform in the same way.

Table 1: Symmetry-compatible optimizer routing for representative neural-network matrix blocks. Each parameter block has a natural symmetry group or quotient structure, which determines the admissible optimizer class. For softmax heads and MoE routers, shared-logit-shift invariance induces a quotient geometry, and centered updates remove the unidentifiable all-ones row direction.

Parameter block	**Symmetry / quotient**	**Compatible update classes**
Linear and attention projections	left/right feature-basis changes, $\mathbb{O}^{d_{\text{out}}} \times \mathbb{O}^{d_{\text{in}}}$	full spectral, polar, or bi-orthogonally equivariant updates
Embedding matrices	token relabeling and feature-basis changes, $\mathbb{P}^v \times \mathbb{O}^d$	LPRO updates, including row-normalized, right-spectral, and hybrid updates
Classification or LM heads	class or token relabeling, feature-basis changes, and shared-logit-shift quotient	LPRO updates and centered variants $\Pi_v^{\perp}\mathcal{V}(\Pi_v^{\perp}D)$
MLP and SwiGLU projections	hidden-unit permutations across coupled gate/up/down projections	row-aware, column-aware, one-sided spectral, or hybrid updates respecting the shared permutation gauge
MoE expert weights	expert relabeling and internal feature geometry	expert-wise permutation-compatible, one-sided spectral, row-normalized, or hybrid updates
MoE routers	expert relabeling and shared-logit-shift quotient	centered row-normalized, centered left-spectral, or centered hybrid updates

5.2 Representative parameter blocks

Embedding matrices and untied classification or language-model heads have rows indexed by tokens or output classes and columns representing latent feature coordinates. Row labels may be permuted but not arbitrarily rotated, while the feature coordinates may be changed by a right orthogonal transformation when adjacent hidden representations are transformed accordingly. Thus their natural symmetry class is left-permutation/right-orthogonal equivariance. This motivates LPRO-compatible updates, including right-spectral maps $\mathcal{U}_{\mathsf{R}}(D) = D\Phi(D^\top D)$, row-normalized maps, and hybrid row-norm/right-spectral maps. For softmax heads, there is also a shared-logit-shift invariance: replacing W_{head} by $W_{\mathsf{head}} + \mathbf{1}_v a^\top$ shifts all class logits by the same scalar and leaves the induced probability vector unchanged. In this quotient geometry, centered updates of the form $\Pi_v^\perp \mathcal{V}(\Pi_v^\perp D)$ are natural.

Feedforward blocks with elementwise nonlinearities have a different symmetry. For a two-layer MLP $x \mapsto W_2\sigma(W_1 x)$, arbitrary rotations of the hidden units do not preserve the function class because σ acts coordinatewise. However, hidden-unit permutations do preserve the block if rows of W_1 and columns of W_2 are permuted consistently. Thus the first projection has a left-permutation/right-orthogonal structure, while the second has the transposed right-permutation/left-orthogonal structure. The same principle applies to gated transformer MLPs such as SwiGLU,

$$x \mapsto W_{\mathsf{down}}\big(\sigma(W_{\mathsf{gate}}x) \odot W_{\mathsf{up}}x\big),$$

where W_{gate}, W_{up}, and W_{down} share the same intermediate hidden-unit permutation gauge. Symmetry-compatible routing should therefore respect the joint permutation of gate, up, and down projections rather than treating the intermediate dimension as fully rotationally invariant.

Mixture-of-experts layers introduce an additional indexed dimension. Expert weights are equivariant under expert relabeling, so update maps should commute with expert permutations. Depending on the internal structure of each expert, one may also impose one-sided feature-space equivariance within the expert matrices. Routers have one further quotient structure: a router matrix $W_{\mathsf{router}} \in \mathbb{R}^{E\times d}$ maps token representations to expert logits, and adding the same row vector to all experts shifts all logits by a token-dependent scalar without changing the softmax probabilities. Hence router updates should be centered with respect to $\Pi_E^\perp = I_E - \frac{1}{E}\mathbf{1}_E\mathbf{1}_E^\top$. Typical choices include centered row-normalized updates and centered left-spectral updates, for example, $\mathcal{U}_{\mathsf{router}}(D) = \Pi_E^\perp \mathcal{V}(\Pi_E^\perp D)$, where $\mathcal{V}$ is row-normalized, left-spectral, or hybrid. The final projection is important because it guarantees that the update contains no component in the shared-logit-shift direction.

5.3 Architecture–optimizer co-design

The examples above show that optimizer design should not be treated as independent of architecture. A fully bi-orthogonal update is natural for a matrix representing a linear operator between Euclidean spaces, but it may be too invariant for a dimension indexing tokens, hidden units, experts, or classes. Conversely, purely coordinatewise updates can be too basis-dependent when a latent feature dimension is identifiable only up to an orthogonal change of basis. Symmetry-compatible optimizer design lies between these extremes: it uses the largest update class compatible with the functional invariances of each parameter block.

This perspective suggests several practical guidelines. Parameter blocks should be grouped according to symmetry type rather than only tensor shape or module name. Tied or jointly parameterized blocks should be optimized using update maps that respect the shared gauge. Centered parameters, such as softmax heads or routers, should include the final centering projection

so that optimization occurs in the identifiable quotient geometry. Finally, normalized matrix updates such as polar, row-normalized, or hybrid maps should be paired with scale-aware normalization, adaptive stepsizes, or inexact-oracle controls to maintain descent.

Thus the resulting optimizer is not a single universal rule applied to all tensors, but a symmetry-compatible collection of matrix-gradient methods matched to the structure of the model: embeddings and heads use LPRO-compatible updates, MLP projections use permutation-compatible coupled updates, MoE experts use expert-permutation-compatible updates, and routers use centered expert-permutation-compatible updates.

6 Numerical experiments

We present two representative pre-training experiments illustrating the symmetry-compatible optimizer-routing principle. The first uses a Gemma 3 1B-style dense transformer and tests updates for large vocabulary-indexed embedding and LM head matrices, as well as tall-skinny SwiGLU MLP projections. The second uses a downsized gpt-oss-style sparse MoE transformer and tests the same principle in the presence of expert weights and softmax router matrices. Additional experiments on Qwen3-0.6B-style dense pre-training and OLMoE-1B-7B-style MoE pre-training are reported in Section D. These experiments are intended as controlled architecture checks for the optimizer-routing principle rather than statistically exhaustive large-scale optimizer benchmarks.

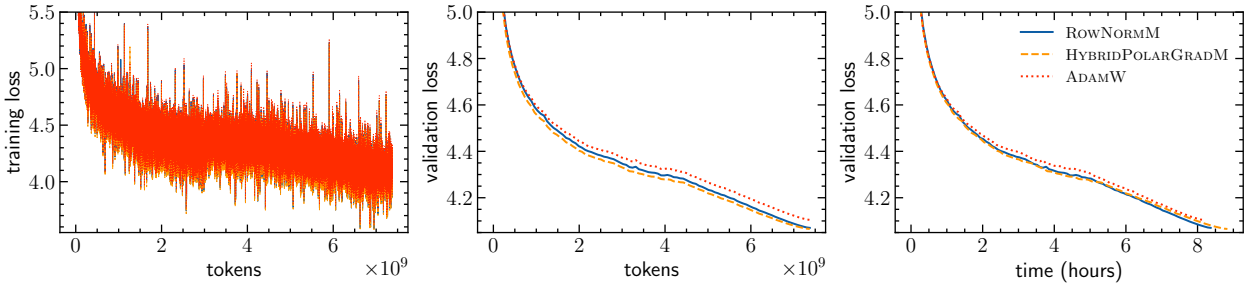
Across experiments, we instantiate a layerwise optimizer stack: linear and attention projections use spectral or head-wise spectral updates; embeddings and LM heads use row-normalized, right-spectral, or hybrid updates; SwiGLU MLP projections use updates matched to intermediate-neuron geometry; MoE routers use centered row-wise or left-spectral updates; and scalar or vector parameters use standard coordinatewise optimizers. We write RowNormM for a row-normalized momentum update and HybridPolarGradM for a hybrid row-norm/right-spectral momentum update. Unless otherwise specified, hidden and attention matrices use Muon [31] with Polar Express coefficients [1], scalar and vector parameters use AdamW [46], and all models are pre-trained on a 10B-token subset of FineWeb-Edu [47] with context length 1024. Further implementation and hyperparameter details are given in Section B.

6.1 Dense transformer pre-training: Gemma 3 1B-style model

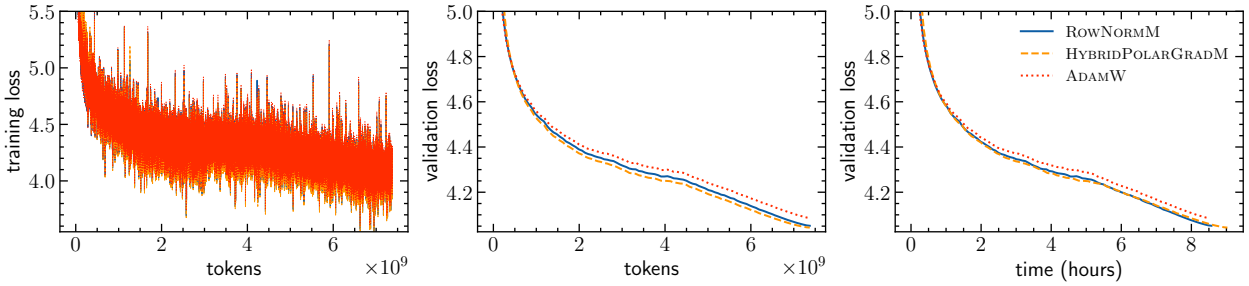
We first pre-train a Gemma 3 1B-style dense language model [20]. The model has vocabulary size 262,144, hidden dimension 1152, and untied input and output embeddings. To keep the model size close to one billion trainable parameters, we reduce the number of hidden layers from 26 to 18, resulting in 1,087,138,944 trainable parameters. This setting is a useful test case for symmetry-compatible updates because the embedding and LM head matrices are large, vocabulary-indexed, and potentially anisotropic.

We compare three optimizer assignments for the input embedding and LM head matrices: RowNormM, HybridPolarGradM, and AdamW. We evaluate these choices under two assignments for the SwiGLU MLP projection matrices: Muon and HybridPolarGradM. All remaining parameter classes use the same optimizer assignments across the six runs.

The final validation losses for configurations (i)–(iii) are 4.0702, 4.0655, and 4.1046 in Figure 1a, and 4.0516, 4.0435, and 4.0862 in Figure 1b. Thus HybridPolarGradM gives the best final validation loss in both MLP settings, while RowNormM also substantially improves over AdamW on the vocabulary-indexed matrices. Replacing Muon by HybridPolarGradM on the SwiGLU MLP projections improves all three embedding/head assignments, reducing the final validation losses by 0.0186, 0.0220, and 0.0184, respectively. This is consistent with the tall-skinny geometry of



(a) SwiGLU MLP projection matrices use Muon, equivalently RightPolarGradM with $\alpha = 0$



(b) SwiGLU MLP projection matrices use HybridPolarGradM with a row-norm/right-spectral composition

Figure 1: Training and validation losses for Gemma 3 1B-style pre-training. In each subfigure, configurations (i)–(iii) differ only in the optimizer assigned to the input embedding and LM head matrices: RowNormM, HybridPolarGradM, or AdamW.

the Gemma 3 1B-style MLP projections, where $d_{\text{model}} = 1152 \ll d_{\text{ff}} = 6912$ and row-scale imbalance across intermediate neurons can matter. The best-performing configuration is therefore the fully hybrid assignment for the embedding, LM head, and SwiGLU MLP matrices, while RowNormM provides a cheaper symmetry-compatible alternative that avoids inverse-square-root computations.

6.2 Mixture-of-experts pre-training: downsized gpt-oss model

We next pre-train a downsized gpt-oss-style sparse MoE model [54]. To obtain a tractable experimental variant, we reduce the number of hidden layers from 24 to 12, the hidden and intermediate dimensions from 2880 to 2048, and the number of experts from 32 to 16. The resulting model has vocabulary size 201,088 and 3,467,779,008 trainable parameters.

We compare four optimizer assignments for the embedding, LM head, and router matrices: (i) RowNormM for all three classes; (ii) RowNormM for the embedding and LM head and LeftPolarGradM for routers; (iii) RowNormM for the embedding and LM head and AdamW for routers; and (iv) AdamW for all three classes. All other parameter classes use the same optimizer assignments: Muon for hidden and attention matrices, geometry-compatible updates for expert SwiGLU tensors, and AdamW for scalar and vector parameters.

The final validation losses for configurations (i)–(iv) are 4.3080, 4.3136, 4.3388, and 4.3687, respectively. The fully coordinatewise baseline in configuration (iv), which uses AdamW for the embedding, LM head, and router matrices, obtains the worst final validation loss. The three configurations using RowNormM for the embedding and LM head matrices all improve substantially over this baseline, suggesting that the benefit of symmetry-compatible updates for vocabulary-indexed matrices persists in sparse MoE training. Among configurations (i)–(iii), the differences are smaller: RowNormM routers perform best, LeftPolarGradM routers are close, and AdamW

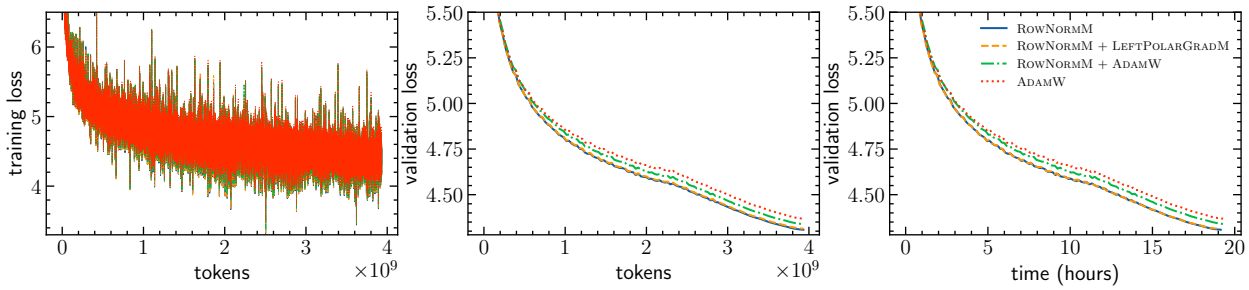


Figure 2: Training and validation losses for downsized gpt-oss pre-training. Configurations (i)–(iv) differ in the optimizers assigned to the embedding, LM head, and router matrices.

routers are slightly worse. This ordering suggests that router geometry can matter, although the dominant improvement in this setting comes from replacing AdamW on the large vocabulary-indexed matrices.

6.3 Summary of empirical findings

The two experiments support the same qualitative conclusion: symmetry-compatible optimizer assignments improve pre-training behavior for matrix blocks whose geometry is not well matched by coordinatewise updates. In the Gemma 3 1B-style dense transformer, replacing AdamW on the embedding and LM head matrices with row-normalized or hybrid updates improves validation loss, and hybrid row-norm/right-spectral updates further improve the tall-skinny SwiGLU MLP projections. In the downsized gpt-oss-style MoE transformer, replacing AdamW on the large vocabulary-indexed matrices again improves validation loss, while geometry-compatible router updates provide additional gains over a coordinatewise router baseline.

Taken together, the experiments support the view that symmetry-compatible optimizer design is most naturally applied as a layerwise optimizer stack rather than as a single global replacement for coordinatewise adaptive gradient methods.

7 Conclusion

This paper developed a symmetry-compatible framework for matrix-gradient methods. The central principle is that an optimizer update should commute with the transformations that preserve the meaning of the matrix variable being optimized. This leads to group-equivariant update maps and provides a coordinate-free way to organize matrix optimization methods beyond coordinatewise preconditioning.

We showed that bi-orthogonal equivariance leads to spectral update maps, while one-sided, permutation, centered, row-normalized, and hybrid equivariance classes arise when only part of the matrix carries Euclidean structure or when rows represent indexed objects such as tokens, classes, hidden units, or experts. These update classes clarify why fully rotational matrix updates are natural for linear-operator blocks but may be too invariant for indexed dimensions, and why coordinatewise updates may be too basis-dependent when latent feature spaces admit orthogonal changes of basis.

The convergence analysis separated structural equivariance from descent. Equivariance determines the admissible form of the update map, while convergence follows from scalar alignment and update-size quantities. This yielded a unified descent principle, sublinear convergence to stationar-

ity, linear convergence under the Polyak–Łojasiewicz condition, and perturbation guarantees for momentum and inexact-oracle variants. The same template applies to spectral, one-sided spectral, row-normalized, centered, and hybrid maps once the corresponding alignment and size bounds are verified.

We also applied the framework to optimizer design for neural network training. Embeddings and classification heads lead to LPRO-compatible updates; MLPs with elementwise nonlinearities require permutation-compatible updates across hidden units; MoE expert weights require expert-permutation-compatible updates; and routers require centered updates because of shared-row-shift invariance. These examples suggest that optimizer design should be co-designed with architecture: when a parameter block has a meaningful symmetry or quotient structure, its update class should be chosen to respect that structure rather than assumed to follow a single universal entrywise rule. The numerical experiments illustrate this principle in dense and sparse language-model pre-training.

Several directions remain open. One is to sharpen the convergence theory for normalized updates, such as polar, row-normalized, and hybrid polar methods, under realistic stochastic, minibatch, and adaptive-stepsize regimes. Another is to develop sharper inexact-oracle analyses for practical matrix computations, including Newton–Schulz iteration, Polar Express [1], QDWH [52], ZOLO-PD [53], and low-precision inverse-square-root routines. A third direction is to extend symmetry-compatible optimizer routing to broader architecture families, including attention variants, low-rank adapters, recurrent state-space models, and structured MoE models. More broadly, the results point toward a systematic theory of architecture-aware first-order optimization, in which the geometry of each parameter block informs the optimizer class used to train it.

Appendix A. Proofs for technical details

This appendix collects proof details and auxiliary calculations used in the convergence analysis. The material is included to keep the main text concise while making the submitted manuscript self-contained.

A.1 Centered quotient geometry

We record the quotient-geometric facts underlying centered updates for softmax heads and MoE routers. Let $W \in \mathbb{R}^{v\times d}$ be a softmax classification matrix. For an input representation $h \in \mathbb{R}^d$, replacing W by $W + \mathbf{1}_v a^\top$ shifts all logits by the same scalar $a^\top h$, and therefore leaves the softmax probabilities unchanged:

$$\operatorname{softmax}\big((W + \mathbf{1}_v a^\top)h\big) = \operatorname{softmax}\big(Wh + \mathbf{1}_v(a^\top h)\big) = \operatorname{softmax}(Wh).$$

Thus the identifiable parameter is an equivalence class in $\mathbb{R}^{v\times d}/(\mathbf{1}_v\mathbb{R}^{1\times d})$. The orthogonal representative is obtained by the centering projector $\Pi_v^\perp := I_v - \frac{1}{v}\mathbf{1}_v\mathbf{1}_v^\top$.

The same structure appears for MoE routers. If $W \in \mathbb{R}^{e\times d}$ maps token representations to expert logits, then W and $W + \mathbf{1}_e a^\top$ induce the same softmax routing probabilities. Hence the meaningful router parameter lies in $\mathbb{R}^{e\times d}/(\mathbf{1}_e\mathbb{R}^{1\times d})$, with centered representative $\Pi_e^\perp W$, where $\Pi_e^\perp := I_e - \frac{1}{e}\mathbf{1}_e\mathbf{1}_e^\top$.

A centered update map is one whose output lies in the centered row subspace. A convenient implementation is $\mathcal{U}_{\mathsf{ctr}}(D) = \Pi^\perp \mathcal{V}(\Pi^\perp D)$, where $\Pi^\perp$ is either $\Pi_v^\perp$ or $\Pi_e^\perp$. The first projection removes the shared-shift component of the input direction, while the final projection ensures that the update remains tangent to the quotient representative. Since $P\Pi^\perp = \Pi^\perp P$ for every compatible permutation matrix P, the centered construction is compatible with row relabeling whenever the base map $\mathcal{V}$ is permutation equivariant.

A.2 Alignment and update-size calculations

We collect the elementary calculations used to verify the alignment and update-size conditions for common update maps.

Spectral maps. Let

$$G = U\,\mathrm{Diag}_{m,n}(\sigma(G))V^\top \quad \text{and} \quad \mathcal{U}(G) = U\,\mathrm{Diag}_{m,n}(\psi(\sigma(G)))V^\top.$$

Orthogonality of U and V gives $\langle\!\langle G, \mathcal{U}(G)\rangle\!\rangle_{\mathrm{F}} = \langle \sigma(G), \psi(\sigma(G))\rangle$ and $|||\mathcal{U}(G)|||_{\mathrm{F}}^2 = \|\psi(\sigma(G))\|_2^2$. Thus the scalar singular-value inequalities in (4.4) imply Assumption 4.3.

One-sided spectral maps. For a right-spectral map $\mathcal{U}_{\mathsf{R}}(G) = G\Phi(G^\top G)$, write $G^\top G = V\,\mathrm{Diag}(t)V^\top$, where $t_i = \lambda_i(G^\top G) = \sigma_i(G)^2$ for $1 \leqslant i \leqslant d$, with zero padding if necessary. If $\Phi(G^\top G) = V\,\mathrm{Diag}(\phi(t))V^\top$, then $\langle\!\langle G, \mathcal{U}_{\mathsf{R}}(G)\rangle\!\rangle_{\mathrm{F}} = \sum_{i=1}^d t_i\phi_i(t)$ and $|||\mathcal{U}_{\mathsf{R}}(G)|||_{\mathrm{F}}^2 = \sum_{i=1}^d t_i\phi_i(t)^2$. The left-spectral case follows by transposition.

Polar and nuclear-scaled polar maps. Let $G = U\,\mathrm{Diag}(\sigma)V^\top$ be a compact singular value decomposition with rank r. The polar update $\mathrm{polar}(G) = UV^\top$ satisfies

$$\langle\!\langle G, \mathrm{polar}(G)\rangle\!\rangle_{\mathrm{F}} = |||G|||_{\mathrm{nuc}}, \qquad |||\mathrm{polar}(G)|||_{\mathrm{F}}^2 = r.$$

Thus polar updates have strong alignment but do not satisfy a uniform update-size bound near $G = 0$. The nuclear-scaled polar update $\mathcal{U}_{\mathsf{nsp}}(G) \coloneqq \frac{|||G|||_{\mathrm{nuc}}}{r}\mathrm{polar}(G)$ satisfies

$$\langle\!\langle G, \mathcal{U}_{\mathsf{nsp}}(G)\rangle\!\rangle_{\mathrm{F}} = \frac{|||G|||_{\mathrm{nuc}}^2}{r}, \qquad |||\mathcal{U}_{\mathsf{nsp}}(G)|||_{\mathrm{F}}^2 = \frac{|||G|||_{\mathrm{nuc}}^2}{r}.$$

Since $|||G|||_{\mathrm{F}} \leqslant |||G|||_{\mathrm{nuc}} \leqslant \sqrt{r}\,|||G|||_{\mathrm{F}}$, this gives $\langle\!\langle G, \mathcal{U}_{\mathsf{nsp}}(G)\rangle\!\rangle_{\mathrm{F}} \geqslant r^{-1}|||G|||_{\mathrm{F}}^2$ and $|||\mathcal{U}_{\mathsf{nsp}}(G)|||_{\mathrm{F}}^2 \leqslant |||G|||_{\mathrm{F}}^2$.

Row-normalized maps. For $\mathcal{U}_{\mathsf{row}}(G) = \mathrm{Diag}(\eta(r_1(G)), \dots, \eta(r_v(G)))G$, where $r_i(G) = \|G_{i,:}\|_2$, one has

$$\langle\!\langle G, \mathcal{U}_{\mathsf{row}}(G)\rangle\!\rangle_{\mathrm{F}} = \sum_{i=1}^v \eta(r_i(G))r_i(G)^2, \qquad |||\mathcal{U}_{\mathsf{row}}(G)|||_{\mathrm{F}}^2 = \sum_{i=1}^v \eta(r_i(G))^2 r_i(G)^2.$$

Exact row normalization has $\langle\!\langle G, \mathcal{U}_{\mathsf{rownorm}}(G)\rangle\!\rangle_{\mathrm{F}} = \sum_i r_i(G)$ and $|||\mathcal{U}_{\mathsf{rownorm}}(G)|||_{\mathrm{F}}^2 = \sharp\{i : r_i(G) > 0\}$, so it is naturally analyzed with adaptive or scale-aware stepsizes.

A scale-aware row-normalized variant is obtained by setting $|||G|||_{2,1} \coloneqq \sum_i \|G_{i,:}\|_2$, $s(G) \coloneqq \sharp\{i : \|G_{i,:}\|_2 > 0\}$, and

$$\mathcal{U}_{\mathsf{row},2,1}(G) \coloneqq \frac{|||G|||_{2,1}}{s(G)}\mathcal{U}_{\mathsf{row},0}(G),$$

where $\mathcal{U}_{\mathsf{row},0}$ is exact row normalization on nonzero rows. Then

$$\langle\!\langle G, \mathcal{U}_{\mathsf{row},2,1}(G)\rangle\!\rangle_{\mathrm{F}} = |||\mathcal{U}_{\mathsf{row},2,1}(G)|||_{\mathrm{F}}^2 = \frac{|||G|||_{2,1}^2}{s(G)}.$$

Since $|||G|||_{\mathrm{F}} \leqslant |||G|||_{2,1} \leqslant \sqrt{s(G)}|||G|||_{\mathrm{F}}$, this update satisfies the alignment and update-size bounds with $\alpha = 1/v$ and $\beta = 1$ uniformly over nonzero G.

Centered maps. Let $\mathcal{U}_{\mathsf{ctr}}(G) = \Pi^{\perp}\mathcal{V}(\Pi^{\perp}G)$. Since $\Pi^{\perp}$ is an orthogonal projector and $\mathcal{U}_{\mathsf{ctr}}(G)$ is centered, we have

$$\langle\!\langle G, \mathcal{U}_{\mathsf{ctr}}(G)\rangle\!\rangle_{\mathrm{F}} = \big\langle\!\big\langle \Pi^{\perp}G, \mathcal{V}(\Pi^{\perp}G)\big\rangle\!\big\rangle_{\mathrm{F}}, \qquad |||\mathcal{U}_{\mathsf{ctr}}(G)|||_{\mathrm{F}} \leqslant |||\mathcal{V}(\Pi^{\perp}G)|||_{\mathrm{F}}.$$

Thus any alignment and update-size bounds for $\mathcal{V}$ on centered directions transfer to $\mathcal{U}_{\mathsf{ctr}}$ with the ambient gradient replaced by $\Pi^{\perp}G$.

Hybrid maps. For hybrid row-norm/right-spectral maps, define $\mathsf{Y}(G) \coloneqq \mathsf{A}(G)G$ and $\mathcal{U}_{\mathsf{hyb}}(G) \coloneqq \mathsf{Y}(G)\Phi(\mathsf{Y}(G)^{\top}\mathsf{Y}(G))$. If $\mathsf{A}(G)$ and $\Phi(\mathsf{Y}(G)^{\top}\mathsf{Y}(G))$ are symmetric positive semidefinite and satisfy $a_{\min}I_v \preccurlyeq \mathsf{A}(G) \preccurlyeq a_{\max}I_v$ and $\rho_{\min}I_d \preccurlyeq \Phi(\mathsf{Y}(G)^{\top}\mathsf{Y}(G)) \preccurlyeq \rho_{\max}I_d$, then

$$\langle\!\langle G, \mathcal{U}_{\mathsf{hyb}}(G)\rangle\!\rangle_{\mathrm{F}} \geqslant a_{\min}\rho_{\min}|||G|||_{\mathrm{F}}^2, \qquad |||\mathcal{U}_{\mathsf{hyb}}(G)|||_{\mathrm{F}}^2 \leqslant a_{\max}^2\rho_{\max}^2|||G|||_{\mathrm{F}}^2.$$

Hence Assumption 4.3 holds with $\alpha = a_{\min}\rho_{\min}$ and $\beta = a_{\max}^2\rho_{\max}^2$.

A.3 Momentum and inexact-oracle details

For a momentum direction M_k, the descent analysis applies with $D_k = M_k$, $a_k = \langle\!\langle \nabla f(W_k), \mathcal{U}(M_k)\rangle\!\rangle_{\mathrm{F}}$, and $b_k = |||\mathcal{U}(M_k)|||_{\mathrm{F}}^2$. If these quantities satisfy the momentum alignment and size bounds in (4.5), then the proof of Theorem 4.1 applies verbatim with α_m, β_m in place of α, β.

For inexact oracles, let $\widehat{\mathcal{U}}(D_k) = \mathcal{U}(D_k) + E_k$. If the ideal update satisfies

$$\langle\!\langle \nabla f(W_k), \mathcal{U}(D_k)\rangle\!\rangle_{\mathrm{F}} \geqslant \alpha|||\nabla f(W_k)|||_{\mathrm{F}}^2, \qquad |||\mathcal{U}(D_k)|||_{\mathrm{F}}^2 \leqslant \beta|||\nabla f(W_k)|||_{\mathrm{F}}^2,$$

and $|||E_k|||_{\mathrm{F}} \leqslant \delta|||\nabla f(W_k)|||_{\mathrm{F}}$ with $0 \leqslant \delta < \alpha$, then Cauchy–Schwarz and the triangle inequality give

$$\big\langle\!\big\langle \nabla f(W_k), \widehat{\mathcal{U}}(D_k)\big\rangle\!\big\rangle_{\mathrm{F}} \geqslant (\alpha - \delta)|||\nabla f(W_k)|||_{\mathrm{F}}^2, \qquad |||\widehat{\mathcal{U}}(D_k)|||_{\mathrm{F}}^2 \leqslant (\sqrt{\beta} + \delta)^2|||\nabla f(W_k)|||_{\mathrm{F}}^2.$$

Therefore, we obtain

$$f(W_{k+1}) \leqslant f(W_k) - \gamma\left(\alpha - \delta - \frac{L\gamma(\sqrt{\beta} + \delta)^2}{2}\right)|||\nabla f(W_k)|||_{\mathrm{F}}^2,$$

and descent holds whenever $0 < \gamma < 2(\alpha - \delta)/(L(\sqrt{\beta} + \delta)^2)$.

For right polar updates, if $\widehat{\mathcal{U}}(D) = D\mathsf{Q}(D)$ approximates $\mathcal{U}(D) = D(D^{\top}D)^{-1/2}$, then

$$|||\widehat{\mathcal{U}}(D) - \mathcal{U}(D)|||_{\mathrm{F}} \leqslant |||D|||_{\mathrm{F}}|||\mathsf{Q}(D) - (D^{\top}D)^{-1/2}|||_{\mathrm{S}}.$$

The analogous left-polar bound is

$$|||\mathsf{P}(D)D - (DD^{\top})^{-1/2}D|||_{\mathrm{F}} \leqslant |||\mathsf{P}(D) - (DD^{\top})^{-1/2}|||_{\mathrm{S}}|||D|||_{\mathrm{F}}.$$

Singular or nearly singular Gram matrices are interpreted through support-restricted, truncated, or damped inverse square roots; the resulting damped or truncated map is then treated as the ideal update $\mathcal{U}$, with remaining numerical error absorbed into E_k.

Appendix B. Optimizer definitions and implementation details

This section summarizes the practical optimizer variants used in Section 6. All matrix-valued variants are applied to momentum directions. For a block W_b with stochastic gradient $G_{b,k}$, we form $M_{b,k} = \beta M_{b,k-1} + (1-\beta)G_{b,k}$ with $M_{b,-1} = 0$, and update $W_{b,k+1} = W_{b,k} - \gamma_{b,k}\mathcal{U}_b(M_{b,k})$.

RowNormM. For $D \in \mathbb{R}^{v\times d}$, we define

$$\mathcal{U}_{\textsc{RowNormM}}(D)_{i,:} \coloneqq \frac{D_{i,:}}{\|D_{i,:}\|_2 + \varepsilon}.$$

For routers, the centered version is $\mathcal{U}_{\textsc{cRowNormM}}(D) \coloneqq \Pi_e^{\perp}\mathcal{U}_{\textsc{RowNormM}}(\Pi_e^{\perp}D)$.

RightPolarGradM and LeftPolarGradM. The right- and left-polar momentum updates are $\mathcal{U}_{\textsc{RightPolarGradM}}(D) \coloneqq D(D^\top D)^{-1/2}$ and $\mathcal{U}_{\textsc{LeftPolarGradM}}(D) \coloneqq (DD^\top)^{-1/2}D$, with inverse square roots taken on the appropriate support. For routers, we use the centered variant

$$\Pi_e^{\perp}\mathcal{U}_{\textsc{LeftPolarGradM}}(\Pi_e^{\perp}D).$$

HybridPolarGradM. For $D \in \mathbb{R}^{v\times d}$, we define

$$\mathsf{A}(D) \coloneqq \mathrm{Diag}\left(\frac{1}{\|D_{1,:}\|_2 + \varepsilon}, \ldots, \frac{1}{\|D_{v,:}\|_2 + \varepsilon}\right), \qquad \mathsf{Y}(D) \coloneqq \mathsf{A}(D)D.$$

The hybrid row-norm/right-polar update is

$$\mathcal{U}_{\textsc{HybridPolarGradM}}(D) \coloneqq \mathsf{Y}(D)\big(\mathsf{Y}(D)^\top \mathsf{Y}(D)\big)^{-1/2}.$$

For matrices whose intermediate-neuron geometry appears along the column axis, we apply the same construction to $D^\top$ and transpose back.

Muon and numerical oracles. For hidden and attention matrices, we use Muon [31] with Polar Express coefficients [1]. For HybridPolarGradM, we use Gram Newton–Schulz iterations [75]. Unless otherwise specified, matrix inverse-square-root or polar computations use 5 inner iterations and $\varepsilon_{\mathrm{NS}} = 10^{-7}$, and row-normalized updates use $\varepsilon = 10^{-8}$. Fused attention and SwiGLU tensors are split or reshaped along their functional axes before applying geometry-aware updates.

Routing. Linear and attention projection matrices use Muon or head-wise Muon. Input embeddings and untied LM heads use RowNormM, HybridPolarGradM, or AdamW depending on the experimental configuration. Dense and expert SwiGLU MLP projections use either Muon or HybridPolarGradM along the intermediate-neuron geometry. MoE routers use RowNormM, LeftPolarGradM, centered variants, or AdamW. Scalar and vector parameters, including biases and normalization parameters, use AdamW.

Appendix C. Model, training, and hyperparameter details

We summarize the model and training configurations used in the experiments. All models are trained on a 10B-token subset of FineWeb-Edu [47] with context length 1024. Unless otherwise specified, we use the same data order, tokenization protocol, validation set, and training schedule across optimizer configurations within each architecture. Matrix weights are initialized with Gaussian entries of mean zero and standard deviation 0.02.

For AdamW on scalar and vector parameters, we use linear warmup followed by cosine decay, with 100 warmup steps and a half-cosine decay to zero. For matrix-valued symmetry-compatible updates, we use a stable-decay schedule: the learning rate is held fixed for the first 60% of training

Table 2: Modified model architectures used in the pre-training experiments.

Model	Qwen3-0.6B	Gemma 3 1B	OLMoE-1B-7B	gpt-oss
Trainable parameters	625,784,832	1,087,138,944	2,824,177,664	3,467,779,008
d_{model}	1024	1152	2048	2048
d_{ff}	3072	6912	1024	2048
n_{layers}	20	18	12	12
n_{heads} (Q / KV)	16/8	4/1	16/16	64/8
d_{head}	128	256	128	64
n_{experts}	N/A	N/A	32	16
Activated experts	N/A	N/A	8	4
Vocabulary size	151,936	262,144	50,304	201,088
Activation	SwiGLU	GeGLU with tanh	SwiGLU	SwiGLU

Table 3: Training configurations.

Model	Qwen3-0.6B	Gemma 3 1B	OLMoE-1B-7B	gpt-oss
Per-device batch size	28	18	20	8
Training steps	30,000	50,000	30,000	60,000
Training tokens	6.88B	7.37B	4.92B	3.93B
Validation tokens	10.55M	10.47M	10.49M	10.49M
Precision	bf16	bf16	bf16	bf16
Data-parallel size	8 H200	8 H200	8 H200	8 H200

steps and then linearly decayed to zero over the remaining 40%. Weight decay is set to zero for embeddings, LM heads, and routers in the reported comparisons, and to 0.001 for most remaining matrix blocks unless otherwise specified. Full per-configuration learning-rate tables are available in the extended arXiv version [37].

Appendix D. Additional experimental results

We report two additional pre-training experiments as compact robustness checks for the optimizer-routing principle. Their details can be found in the extended arXiv version [37].

Qwen3-0.6B-style dense pre-training. We repeat the dense-transformer comparison on a Qwen3-0.6B-style model [59]. The qualitative ordering is consistent with the Gemma 3 1B-style experiment: replacing AdamW on vocabulary-indexed embedding and LM head matrices by row-normalized or hybrid symmetry-compatible updates improves final validation loss, and applying the hybrid row-norm/right-spectral update to SwiGLU MLP projections gives a further but smaller improvement. The final validation losses for configurations (i)–(iii) are 4.2017, 4.2050, and 4.2084 when MLP projections use Muon, and 4.1950, 4.1955, and 4.2046 when MLP projections use HybridPolarGradM.

OLMoE-1B-7B-style sparse MoE pre-training. We also evaluate a sparse MoE setting based on an OLMoE-1B-7B-style architecture [51]. The final validation losses for configurations (i)–(iv) are 4.0955, 4.0815, 4.1187, and 4.1240, respectively. As in the downsized gpt-oss experiment, replacing AdamW on vocabulary-indexed embedding and LM head matrices improves final validation loss. In this setting, the centered left-polar router update gives the best final validation loss, suggesting that router geometry can become more important depending on the MoE architecture and training

configuration.

Table 4: Final validation losses for additional experiments.

Experiment	Final validation losses
Qwen3, MLP uses Muon	(*i*) 4.2017, (*ii*) 4.2050, (*iii*) 4.2084
Qwen3, MLP uses HybridPolarGradM	(*i*) 4.1950, (*ii*) 4.1955, (*iii*) 4.2046
OLMoE-1B-7B-style	(*i*) 4.0955, (*ii*) 4.0815, (*iii*) 4.1187, (*iv*) 4.1240

Overall, these additional experiments support the same conclusion as the main experiments: symmetry-compatible optimizer routing is most useful for matrix blocks with nontrivial indexing, quotient, or feature-space geometry, and its benefits are more naturally realized as a layerwise optimizer stack than as a single global optimizer replacement.

Acknowledgments

This work was supported by computational resources from Prime Intellect.

References

[1] N. Amsel, D. Persson, C. Musco, and R. M. Gower, *The Polar Express: Optimal matrix sign methods and their application to the Muon algorithm*, in International Conference on Learning Representations (ICLR), 2026.

[2] R. Anil, V. Gupta, T. Koren, K. Regan, and Y. Singer, *Scalable second order optimization for deep learning*, Feb 2020, https://arxiv.org/abs/2002.09018.

[3] J. Bernstein, *Modular manifolds*. Thinking Machines Lab: Connectionism, Sep 2025, https://thinkingmachines.ai/blog/modular-manifolds/ (accessed 2026-07-21).

[4] R. Bhatia, *Matrix Analysis*, vol. 169 of Graduate Texts in Mathematics, Springer Science & Business Media, 2013, https://doi.org/10.1007/978-1-4612-0653-8.

[5] S. Buchanan, *A faster manifold Muon with ADMM*. Weblog, Oct 2025, https://sdbuchanan.com/blog/manifold-muon/ (accessed 2026-07-21).

[6] D. Carlson, V. Cevher, and L. Carin, *Stochastic spectral descent for restricted Boltzmann machines*, in Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), 2015.

[7] D. Carlson, E. Collins, Y.-P. Hsieh, L. Carin, and V. Cevher, *Preconditioned spectral descent for deep learning*, in Advances in Neural Information Processing Systems (NeurIPS), 2015.

[8] D. Carlson, Y.-P. Hsieh, E. Collins, L. Carin, and V. Cevher, *Stochastic spectral descent for discrete graphical models*, IEEE Journal of Selected Topics in Signal Processing, 10 (2016), pp. 296–311, https://doi.org/10.1109/JSTSP.2015.2505684.

[9] D. Chang, Y. Liu, and G. Yuan, *On the convergence of Muon and beyond*, Sep 2025, https://arxiv.org/abs/2509.15816.

[10] L. Chen, J. Li, and Q. Liu, *Muon optimizes under spectral norm constraints*, Transactions on Machine Learning Research, (2026), https://openreview.net/forum?id=Blz4hjxLwU.

[11] M. Crawshaw, C. Modi, M. Liu, and R. M. Gower, *An exploration of non-Euclidean gradient descent: Muon and its many variants*, in Proceedings of the International Conference on Machine Learning (ICML), 2026.

[12] D. Davis and D. Drusvyatskiy, *When do spectral gradient updates help in deep learning?*, Dec 2025, https://arxiv.org/abs/2512.04299.

[13] C. Ding, D. Sun, J. Sun, and K.-C. Toh, *Spectral operators of matrices*, Mathematical Programming, 168 (2018), pp. 509–531, https://doi.org/10.1007/s10107-017-1162-3.

[14] C. Ding, D. Sun, J. Sun, and K.-C. Toh, *Spectral operators of matrices: Semismoothness and characterizations of the generalized Jacobian*, SIAM Journal on Optimization, 30 (2020), pp. 630–659, https://doi.org/10.1137/18M1222235.

[15] Z. Du and W. Su, *The Newton–Muon optimizer*, Apr 2026, https://arxiv.org/abs/2604.01472.

[16] J. Duchi, E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research, 12 (2011), pp. 2121–2159.

[17] R. Eschenhagen, A. Cai, T.-H. Lee, and H.-J. M. Shi, *Clarifying Shampoo: Adapting spectral descent to stochasticity and the parameter trajectory*, Feb 2026, https://arxiv.org/abs/2602.09314.

[18] R. Eschenhagen, A. Immer, R. Turner, F. Schneider, and P. Hennig, *Kronecker-factored approximate curvature for modern neural network architectures*, in Advances in Neural Information Processing Systems (NeurIPS), 2023.

[19] K. Frans, S. Levine, and P. Abbeel, *A stable whitening optimizer for efficient neural network training*, in Advances in Neural Information Processing Systems (NeurIPS), 2025.

[20] Gemma Team, *Gemma 3 technical report*, Mar 2025, https://arxiv.org/abs/2503.19786.

[21] D. Goldfarb, Y. Ren, and A. Bahamou, *Practical quasi-Newton methods for training deep neural networks*, in Advances in Neural Information Processing Systems (NeurIPS), 2020.

[22] W. Gong, J. Zazo, Q. Luo, P. Wang, J. Hensman, and C. Ma, *ARO: A new lens on matrix optimization for large models*, Feb 2026, https://arxiv.org/abs/2602.09006.

[23] A. Gonon, A.-A. Muşat, and N. Boumal, *Insights on Muon from simple quadratics*, Feb 2026, https://arxiv.org/abs/2602.11948.

[24] Y. Gu and Z. Xie, *Mano: Restriking manifold optimization for LLM training*, Jan 2026, https://arxiv.org/abs/2601.23000.

[25] V. Gupta, T. Koren, and Y. Singer, *Shampoo: Preconditioned stochastic tensor optimization*, in Proceedings of the International Conference on Machine Learning (ICML), 2018.

[26] N. J. Higham, *Functions of Matrices: Theory and Computation*, Society for Industrial and Applied Mathematics, 2008, https://doi.org/10.1137/1.9780898717778.

[27] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, 1994, https://doi.org/10.1017/CBO9780511840371.

[28] F. Huang, Y. Luo, and S. Chen, *LiMuon: Light and fast Muon optimizer for large models*, in Proceedings of the International Conference on Machine Learning (ICML), 2026.

[29] R. Jiang, Z. Mhammedi, M. Mohri, and A. Mokhtari, *Adaptive matrix online learning through smoothing with guarantees for nonsmooth nonconvex optimization*, in Proceedings of the Conference on Learning Theory (COLT), 2026.

[30] K. Jordan, J. Bernstein, B. Rappazzo, @fernbear.bsky.social, B. Vlado, Y. Jiacheng, F. Cesista, B. Koszarsky, and @Grad62304977, `modded-nanogpt`*: Speedrunning the NanoGPT baseline*, 2024, https://github.com/KellerJordan/modded-nanogpt.

[31] K. Jordan, Y. Jin, V. Boza, Y. Jiacheng, F. Cecista, L. Newhouse, and J. Bernstein, *Muon: An optimizer for hidden layers in neural networks*. Weblog, Dec 2024, https://kellerjordan.github.io/posts/muon/ (accessed 2026-07-21).

[32] G. Y. Kim and M.-h. Oh, *Convergence of Muon with Newton-Schulz*, in International Conference on Learning Representations (ICLR), 2026.

[33] D. P. Kingma and J. L. Ba, *Adam: a method for stochastic optimization*, in International Conference on Learning Representations (ICLR), 2015.

[34] D. Kovalev and E. Borodich, *Non-Euclidean SGD for structured optimization: Unified analysis and improved rates*, Nov 2025, https://arxiv.org/abs/2511.11466.

[35] A. Kravatskiy, I. Kozyrev, N. Kozlov, A. Vinogradov, D. Merkulov, and I. Oseledets, *The Ky Fan norms and beyond: Dual norms and combinations for matrix optimization*, Dec 2025, https://arxiv.org/abs/2512.09678.

[36] T. T.-K. Lau, Q. Long, and W. Su, *PolarGrad: A class of matrix-gradient optimizers from a unifying preconditioning perspective*, May 2025, https://arxiv.org/abs/2505.21799.

[37] T. T.-K. Lau and W. Su, *Symmetry-compatible principle for optimizer design: Embeddings, LM heads, SwiGLU MLPs, and MoE routers*, May 2026, https://arxiv.org/abs/2605.18106.

[38] A. S. Lewis, *The convex analysis of unitarily invariant matrix functions*, Journal of Convex Analysis, 2 (1995), pp. 173–183, https://www.heldermann-verlag.de/jca/jca02/jca02012.pdf.

[39] A. S. Lewis, *Group invariance and convex matrix analysis*, SIAM Journal on Matrix Analysis and Applications, 17 (1996), pp. 927–949, https://doi.org/10.1137/S0895479895283173.

[40] A. S. Lewis, *The mathematics of eigenvalue optimization*, Mathematical Programming, 97 (2003), pp. 155–176, https://doi.org/10.1007/s10107-003-0441-3.

[41] A. S. Lewis and M. L. Overton, *Eigenvalue optimization*, Acta Numerica, 5 (1996), pp. 149–190, https://doi.org/10.1017/S0962492900002646.

[42] J. Li and M. Hong, *A note on the convergence of Muon*, Feb 2025, https://arxiv.org/abs/2502.02900.

[43] X.-L. Li, *Preconditioned stochastic gradient descent*, IEEE Transactions on Neural Networks and Learning Systems, 29 (2017), pp. 1454–1466, https://doi.org/10.1109/TNNLS.2017.2672978.

[44] W. Lin, S. C. Lowe, F. Dangel, R. Eschenhagen, Z. Xu, and R. B. Grosse, *Understanding and improving Shampoo and SOAP via Kullback–Leibler minimization*, in International Conference on Learning Representations (ICLR), 2026.

[45] Y. Liu, A. Yuan, and Q. Gu, *MARS-M: When variance reduction meets matrices*, Oct 2025, https://arxiv.org/abs/2510.21800.

[46] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, in International Conference on Learning Representations (ICLR), 2019.

[47] A. Lozhkov, L. Ben Allal, L. von Werra, and T. Wolf, *FineWeb-Edu: the finest collection of educational content*, 2024, https://doi.org/10.57967/hf/2497.

[48] J. Ma, Y. Huang, Y. Chi, and Y. Chen, *Preconditioning benefits of spectral orthogonalization in Muon*, Jan 2026, https://arxiv.org/abs/2601.13474.

[49] J. Martens and R. Grosse, *Optimizing neural networks with Kronecker-factored approximate curvature*, in Proceedings of the International Conference on Machine Learning (ICML), 2015.

[50] H. B. McMahan and M. Streeter, *Adaptive bound optimization for online convex optimization*, in Proceedings of the Conference on Learning Theory (COLT), 2010.

[51] N. Muennighoff, L. Soldaini, D. Groeneveld, K. Lo, J. Morrison, S. Min, W. Shi, P. Walsh, O. Tafjord, N. Lambert, Y. Gu, S. Arora, A. Bhagia, D. Schwenk, D. Wadden, A. Wettig, B. Hui, T. Dettmers, D. Kiela, A. Farhadi, N. A. Smith, P. W. Koh, A. Singh, and H. Hajishirzi, *OLMoE: Open mixture-of-experts language models*, in International Conference on Learning Representations (ICLR), 2025.

[52] Y. Nakatsukasa, Z. Bai, and F. Gygi, *Optimizing Halley's iteration for computing the matrix polar decomposition*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 2700–2720, https://doi.org/10.1137/090774999.

[53] Y. Nakatsukasa and R. W. Freund, *Computing fundamental matrix decompositions accurately via the matrix sign function in two iterations: The power of Zolotarev's functions*, SIAM Review, 58 (2016), pp. 461–493, https://doi.org/10.1137/140990334.

[54] OpenAI, *gpt-oss-120b & gpt-oss-20b model card*, Aug 2025, https://arxiv.org/abs/2508.10925.

[55] T. Pethick, K. Antonakopoulos, A. Silveti-Falls, L. C. Vankadara, and V. Cevher, *Training neural networks at any scale: An exposition*, IEEE Signal Processing Magazine, 43 (2026), pp. 21–36, https://doi.org/10.1109/MSP.2026.3667310.

[56] T. Pethick, W. Xie, K. Antonakopoulos, Z. Zhu, A. Silveti-Falls, and V. Cevher, *Training deep learning models with norm-constrained LMOs*, in Proceedings of the International Conference on Machine Learning (ICML), 2025.

[57] O. Pooladzandi and X.-L. Li, *Curvature-informed SGD via general purpose Lie-group preconditioners*, Feb 2024, https://arxiv.org/abs/2402.04553.

[58] X. Qi, M. Chen, J. Ye, Y. He, and R. Xiao, *Delving into Muon and beyond: Deep analysis and extensions*, in Proceedings of the International Conference on Machine Learning (ICML), 2026.

[59] Qwen Team, *Qwen3 technical report*, May 2025, https://arxiv.org/abs/2505.09388.

[60] N. Shazeer and M. Stern, *Adafactor: Adaptive learning rates with sublinear memory cost*, in Proceedings of the International Conference on Machine Learning (ICML), 2018.

[61] W. Shen, R. Huang, M. Huang, C. Shen, and J. Zhang, *On the convergence analysis of Muon*, May 2025, https://arxiv.org/abs/2505.23737.

[62] H.-J. M. Shi, T.-H. Lee, S. Iwasaki, J. Gallego-Posada, Z. Li, K. Rangadurai, D. Mudigere, and M. Rabbat, *A distributed data-parallel PyTorch implementation of the distributed Shampoo optimizer for training neural networks at-scale*, Sep 2023, https://arxiv.org/abs/2309.06497.

[63] C. Si, D. Zhang, and W. Shen, *AdaMuon: Adaptive Muon optimizer*, Jul 2025, https://arxiv.org/abs/2507.11005.

[64] D. Su, A. Gu, J. Xu, Y. Tian, and J. Zhao, *GaLore 2: Large-scale LLM pre-training by gradient low-rank projection*, Apr 2025, https://arxiv.org/abs/2504.20437.

[65] W. Su, *Isotropic curvature model for understanding deep learning optimization: Is gradient orthogonalization optimal?*, Nov 2025, https://arxiv.org/abs/2511.00674.

[66] D. Sun and J. Sun, *Löwner's operator and spectral functions in Euclidean Jordan algebras*, Mathematics of Operations Research, 33 (2008), pp. 421–445, https://doi.org/10.1287/moor.1070.0300.

[67] T. Tieleman and G. Hinton, *Lecture 6.5—RMSProp: Divide the gradient by a running average of its recent magnitude*. Coursera: Neural Networks for Machine Learning, 2012, https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf (accessed 2026-07-21).

[68] A. Veprikov, A. Bolatov, S. Horváth, A. Beznosikov, M. Takáč, and S. Hanzely, *Preconditioned norms: A unified framework for steepest descent, quasi-Newton and adaptive methods*, Oct 2025, https://arxiv.org/abs/2510.10777.

[69] N. Vyas, D. Morwani, R. Zhao, I. Shapira, D. Brandfonbrener, L. Janson, and S. Kakade, *SOAP: Improving and stabilizing Shampoo using Adam*, in International Conference on Learning Representations (ICLR), 2025.

[70] T. Xie, H. Luo, H. Tang, Y. Hu, J. K. Liu, Q. Ren, Y. Wang, W. X. Zhao, R. Yan, B. Su, C. Luo, and B. Guo, *Controlled LLM training on spectral sphere*, Jan 2026, https://arxiv.org/abs/2601.08393.

[71] C. Xu, W. Yan, and Y.-J. A. Zhang, *FISMO: Fisher-structured momentum-orthogonalized optimizer*, Jan 2026, https://arxiv.org/abs/2601.21750.

[72] R. Xu, J. Li, and Y. Lu, *On the width scaling of neural optimizers under matrix operator norms I: Row/column normalization and hyperparameter transfer*, Mar 2026, https://arxiv.org/abs/2603.09952.

[73] K. Yang and L. Lai, *Manifold constrained steepest descent*, Jan 2026, https://arxiv.org/abs/2601.21487.

[74] H. Yuan, Y. Liu, S. Wu, X. Zhou, and Q. Gu, *MARS: Unleashing the power of variance reduction for training large models*, in Proceedings of the International Conference on Machine Learning (ICML), 2025.

[75] J. Zhang, N. Amsel, B. Chen, and T. Dao, *Gram Newton-Schulz*. Weblog, Mar 2026, https://dao-ailab.github.io/blog/2026/gram-newton-schulz/ (accessed 2026-07-21).

[76] M. Zhang, Y. Liu, and H. Schaeffer, *AdaGrad meets Muon: Adaptive stepsizes for orthogonal updates*, Sep 2025, https://arxiv.org/abs/2509.02981.

[77] Y. Zhang, S. Xing, J. Huang, K. Lv, Y. Zhou, X. Qiu, Q. Guo, and K. Chen, *Mousse: Rectifying the geometry of Muon with curvature-aware preconditioning*, Mar 2026, https://arxiv.org/abs/2603.09697.

[78] J. Zhao, Z. Zhang, B. Chen, Z. Wang, A. Anandkumar, and Y. Tian, *GaLore: Memory-efficient LLM training by gradient low-rank projection*, in Proceedings of the International Conference on Machine Learning (ICML), 2024.