BiMuon: A GEOMETRY-AWARE STOCHASTIC BILEVEL OPTIMIZATION APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

Stochastic bilevel optimization provides a general framework for many machine learning applications, such as hyperparameter learning, data reweighting, language model alignment, and actor-critic reinforcement learning. However, most existing bilevel optimization methods treat matrix-valued gradients as flattened vectors and perform optimization in the resulting vectorized space, ignoring the matrix structures common in modern neural networks. Moreover, the lower-level (LL) problem curvature can induce significant spectral imbalance in the upper-level (UL) geometry. Such limitations of existing bilevel optimization methods motivate us to develop BiMuon, a geometry-aware stochastic bilevel optimization method that integrates implicit hypergradient estimation in bilevel optimization with Muon-style orthogonalized updates at the UL. To our knowledge, BiMuon is the first stochastic BLO method with Muon-style orthogonalized matrix updates and an $O(\epsilon^{-4})$ iteration-complexity guarantee, matching the best-known rates achieved by Muon-type methods based on matrix parameterizations. To improve computational and parallel efficiency for models with multiple matrix parameters, we further study BiMuon for blockwide diagonal matrix approximation and derive a relative stepsize allocation rule based on UL progress and blockwise LL sensitivity. We evaluate BiMuon on representation learning and blockwise bilevel optimization, providing empirical support for our methods.

1 INTRODUCTION

1) Background and Motivation Stochastic bilevel optimization (BLO) is a unified foundational optimization framework for many machine learning applications. Representative examples include reinforcement learning from human feedback (RLHF) (Chakraborty et al., 2023), data reweighting for large-scale pretraining (Kang et al., 2025), preference-based fine-tuning (Ding et al., 2024), and hyperparameter optimization (Franceschi et al., 2018; Lorraine et al., 2020). These problems naturally admit a BLO formulation, where an upper-level (UL) problem governs a set of control variables, while a lower-level (LL) problem defines the training dynamics of the base model.

In recent years, significant progress has been made in BLO, with algorithms based on implicit differentiation, Neumann-series approximations, and stochastic optimization achieving provable convergence guarantees under nonconvex–strongly-convex assumptions (Franceschi et al., 2018; Lorraine et al., 2020; Ji et al., 2021; Chen et al., 2022; Dagréou et al., 2022). However, almost all BLO methods adopt a flattened-vector-based (FVB) approach to optimize model parameters, vectorizing weight matrices and applying coordinate-wise updates. While the FVB approach is mathematically convenient, it ignores the matrix structures that are ubiquitous in neural network models.

Meanwhile, in contemporary neural architecture training with BLO (e.g., large language models and many Transformer-based architectures), the UL and LL variables can naturally be matrix-based linear operators whose training dynamics exhibit strong spectral anisotropy and low-rank structure (Liu et al., 2025; Zhao et al., 2024). Empirical and theoretical studies consistently show that gradients and update matrices in such models can be highly *ill-conditioned,* with learning dynamics governed by a small number of dominant singular directions (Jordan et al., 2024; Zhao et al., 2024). As a result, FVB updates tend to over-emphasize directions associated with large singular values, while making limited progress along flatter or smaller spectral directions. In BLO, this mismatch can be further execerbated by the LL curvature: as we show theoretically in this paper, poor LL con-

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

ditioning can slow LL optimization and, through the bilevel composition, induce stronger spectral imbalance in the UL geometry and hypergradient.

Interestingly, recent advances in matrix-based optimization based on singular-spectrum regularization, most notably Muon (Jordan et al., 2024) and its variants, demonstrate that orthogonalized updates can better *align optimization dynamics with the spectral structure of neural network matrix parameters* and achieve strong empirical performance (Liu et al., 2025). Specifically, by normalizing singular values, Muon mitigates the dominance of large spectral components and promotes balanced updates across subspaces. Moreover, recent theoretical results have begun to establish convergence guarantees for Muon-style orthogonalized updates in single-level stochastic optimization (Shen et al., 2025; Kim & Oh, 2026). These recent advances in singular-spectrum-based optimizers naturally raise the following fundamental question:

(Q): Can we develop a Muon-style matrix-based algorithm for geometry-aware stochastic bilevel optimization with provable convergence guarantees?

2) Technical Challenges and Major Contributions: Unfortunately, developing singular-spectrum-regularized approaches for BLO is highly non-trivial due to several technical challenges unseen in single-level settings. In BLO, hypergradient computation involves LL Hessian inversion, and LL and UL variables are coupled across iterations. Specifically, each UL update changes the UL variable, thus altering the LL solution that must be tracked by a finite LL optimization process. The resulting LL tracking error in turn affects the subsequent hypergradient estimate, creating feedback loops between matrix-based UL and LL updates (Ji et al., 2021; Grazzi et al., 2021). Moreover, practical Muon implementations often use a finite Newton-Schulz (NS) approximation instead of the exact polar decomposition, incurring an additional layer of approximation errors (Kim & Oh, 2026).

In this work, we address the aforementioned challenges by proposing BiMuon, which, to our knowledge, is the first geometry-aware, matrix-based stochastic BLO algorithm built on Muon-style orthogonalized updates. Our key idea is to decouple bilevel hypergradient estimation from orthogonalized UL updates: the LL problem is first tracked using warm-started stochastic gradient descent (SGD), while the UL hypergradient is estimated using a truncated Neumann-series approximation; orthogonalized updates are then applied at the final UL parameter update to exploit matrix geometry. This design allows BiMuon to retain the standard stochastic bilevel estimation procedure, while enabling matrix-based optimization principles within bilevel learning. Our contributions are threefold:

- We first theoretically show that LL curvature in BLO can induce significant *spectral imbalance* in the UL geometry, highlighting the compelling need for geometry-aware optimization in BLO. To mitigate this spectral imbalance, we propose BiMuon, the first geometry-aware stochastic BLO method with Muon-style orthogonalized matrix updates.
- We then establish the finite-time convergence rate guarantees for both the exact-polar and finite-NS versions of BiMuon. Specifically, both exact-polar and finite-NS versions of BiMuon achieve a finite-time convergence rate of $\mathbb{E}\|\nabla\Phi(X_R)\|_* = O(T^{-1/4})$, where T is the maximum number of iterations. For any given error tolerance $\epsilon > 0$, this converge rate implies an $\widetilde{O}(\epsilon^{-4})$ sample complexity, matching the best-known results for stochastic single-level Muon-type optimization.
- To improve computational and parallel efficiency for models with multiple matrix parameters, we further study BiMuon with blockwise diagonal matrix approximation and propose a step-size allocation rule based on UL progress and blockwise LL sensitivity. We further evaluate BiMuon on representation learning and blockwise bilevel optimization problems, which empirically verify the performance of our proposed BiMuon algorithmic designs.

2 Related Work

In this section, we provide an overview on two lines of research closely related to our work.

1) Stochastic Bilevel Optimization. Gradient-based BLO methods commonly use implicit or iterative differentiation to approximate hypergradients under the strongly convex LL problem assumption (Franceschi et al., 2018; Ghadimi & Wang, 2018; Ji et al., 2021; Yang et al., 2021). Recent stochastic methods further improve LL tracking and hypergradient estimation through single-timescale and variance-reduced schemes (Chen et al., 2022; Dagréou et al., 2022; Chu et al., 2024).

These works primarily study how to estimate and track the hypergradient, while the resulting UL update typically remains flattened-vector-based (FVB). Bregman and Riemannian BLO methods introduce alternative optimization geometries (Huang et al., 2022; Han et al., 2024; Li & Ma, 2025). In contrast, our BiMuon method targets matrix-valued UL variables and exploits the singular-spectrum structure of their hypergradients through Muon-style orthogonalized updates.

2) Muon and Other Matrix-Aware Optimizers. Matrix-aware optimizers exploit structure beyond flattened parameter vectors. For example, Shampoo and SOAP use matrix preconditioning (Gupta et al., 2018; Vyas et al., 2025), while GaLore exploits low-rank gradient structure (Zhao et al., 2024). Most notably, Muon orthogonalizes gradient or momentum matrices by normalizing their singular spectrum (Jordan et al., 2024). Recent works on Muon have verified large-scale empirical performance (Liu et al., 2025) and established theoretical convergence guarantees for exact and finite-NS updates (Shen et al., 2025; Kim & Oh, 2026). Norm-based steepest descent and Scion further connect matrix updates to norm geometry (Bernstein & Newhouse, 2024; Pethick et al., 2025). However, all of these results focus on single-level optimization and do not address LL tracking or implicit hypergradient errors. Blockwise matrix methods motivate processing large parameter matrices separately (Amari et al., 2019; Mendler-Dünner & Lucchi, 2020; Bahamou et al., 2023); our blockwise design additionally uses LL sensitivity to allocate relative stepsizes across blocks.

3 Problem Statement and Preliminaries

In this section, we first review the standard FVB formulation of BLO, which is followed by an in-depth analysis on the significant bilevel-inherent spectral imbalance to motivate our subsequent BiMuon algorithmic design. For matrices in this paper, we let $\langle A, B\rangle = \mathrm{tr}(A^\top B)$, $\|A\|_F = \sqrt{\langle A, A\rangle}$, and let $\|A\|_*$ and $\|A\|_{\mathrm{op}}$ denote the nuclear and spectral norms. For matrix pairs, $\|(X, Y)\|_F^2 := \|X\|_F^2 + \|Y\|_F^2$. For an $m \times n$ UL matrix, $r = \min\{m, n\}$ is the rank cap.

3.1 BLO Problem Statement and the FVB Hypergradient-Based Approaches

1) Problem Statement. In the FVB form, a stochastic BLO problem can generally be written as

$$\begin{aligned} \min_{x\in\mathbb{R}^{d_x}} \quad & \Phi(x) := f(x, y^\star(x)) = \mathbb{E}_\xi[F(x, y^\star(x); \xi)] \\ \text{s.t.} \quad & y^\star(x) \in \arg\min_{y\in\mathbb{R}^{d_y}} g(x, y) := \mathbb{E}_\zeta[G(x, y; \zeta)]. \end{aligned}$$

Here, x denotes the UL variable and $y^\star(x)$ is an optimizer of the LL objective $g(x, \cdot)$ with x being fixed. We follow the standard BLO setting in the literature, where the UL objective Φ is nonconvex in x and the LL objective $g(x, \cdot)$ is μ-strongly convex for some $\mu > 0$, ensuring that the LL solution $y^\star(x)$ is uniquely defined. The expectations in the UL and LL objectives are taken over the distributions of random variables ξ and ζ, respectively.

2) The FVB-Based Hypergradient Approach for BLO. When the LL problem is strongly convex, under the differentiability assumptions (cf. Section 5), the UL objective $\Phi(x) = f(x, y^\star(x))$ admits a well-defined hypergradient. Specifically, by the implicit function theorem (Ghadimi & Wang, 2018), the hypergradient of Φ can be written as:

$$\nabla\Phi(x) = \nabla_x f(x, y^\star(x)) - \nabla^2_{xy} g(x, y^\star(x)) \big[\nabla^2_{yy} g(x, y^\star(x))\big]^{-1} \nabla_y f(x, y^\star(x)). \tag{1}$$

However, direct computation of the inverse-Hessian action in (1) is generally expensive. A standard approach in stochastic BLO is to use the truncated Neumann-series approximation (Ji et al., 2021; Yang et al., 2021) $H^{-1} \approx \eta \sum_{i=0}^{Q} (I - \eta H)^i$, where $H = \nabla^2_{yy} g(x, y)$ is the LL Hessian at the current point and $\eta > 0$ is chosen such that $\|I - \eta H\|_{\mathrm{op}} < 1$. This approximation can be evaluated using Hessian–vector products without forming the Hessian inverse explicitly. We use this standard construction for the hypergradient estimator in Section 4.2.

The FVB formulation can directly extend to matrix-valued variables $X \in \mathbb{R}^{m\times n}$ and $Y \in \mathbb{R}^{h\times s}$ via $x = \mathrm{vec}(X)$ and $y = \mathrm{vec}(Y)$, with $d_x = mn$ and $d_y = hs$, provided the updates remain entry-wise. In matrix form, $\nabla^2_{XY} g$ is the adjoint of $D_X(\nabla_Y g)$ and maps LL matrices to UL matrices; however, such a matrix-valued extension does *not* leverage matrix structures (e.g., spectral properties or low-rank geometry) that arise in modern neural networks. We next use a quadratic bilevel example to show how LL curvature can inherently induce *signifcant spectral imbalance* in the UL geometry.

3.2 Bilevel-Inherent Spectral Imbalance: A Quadratic Case Study

Motivated by the fact that many nonlinear functions of practical interest can be locally well approximated by quadratic functions, we use a quadratic bilevel problem to illustrate how differences in LL curvature can be *amplified* through bilevel composition, resulting in highly unequal singular values in the UL hypergradient. Let $(Z_{\mathrm{tr}}, R_{\mathrm{tr}})$ and $(Z_{\mathrm{val}}, R_{\mathrm{val}})$ denote the training and validation data with n_{tr} and n_{val} samples, respectively. For matrix variables $X, Y \in \mathbb{R}^{m\times n}$, consider the following quadratic UL and LL functions:

$$f(X,Y) = \frac{\|Z_{\mathrm{val}}Y - R_{\mathrm{val}}\|_{\mathrm{F}}^2}{2n_{\mathrm{val}}} + \frac{\gamma}{2}\|X - X_{\mathrm{prior}}\|_{\mathrm{F}}^2, \quad g(X,Y) = \frac{\|Z_{\mathrm{tr}}Y - R_{\mathrm{tr}}\|_{\mathrm{F}}^2}{2n_{\mathrm{tr}}} + \frac{\mu}{2}\|Y - X\|_{\mathrm{F}}^2, \quad (2)$$

where $\mu > 0$ controls the coupling between the UL and LL variables, $\gamma \geq 0$ is the UL regularization parameter, and X_{prior} is a reference matrix. The LL problem adapts Y using the training data while keeping it close to X, whereas the UL objective evaluates the adapted model on validation data.

Define the Hessian matrix that characterizes LL curvature as $H_{\mathrm{LL}} := Z_{\mathrm{tr}}^\top Z_{\mathrm{tr}}/n_{\mathrm{tr}} + \mu I$. Under isotropic validation curvature $Z_{\mathrm{val}}^\top Z_{\mathrm{val}}/n_{\mathrm{val}} = I$ and $\gamma = 0$, it can be readily verified that the induced UL Hessian operator $\mathcal{H}_{\mathrm{UL}} := \nabla^2\Phi(X)$ satisfies:

$$\mathcal{H}_{\mathrm{UL}}[\Delta X] = \mu^2 H_{\mathrm{LL}}^{-2}\Delta X, \qquad \kappa(\mathcal{H}_{\mathrm{UL}}) = \kappa(H_{\mathrm{LL}})^2,$$

where $\kappa(\cdot)$ is the condition number (i.e., the ratio of the largest to the smallest eigenvalue). Thus, curvature differences across LL directions are amplified **quadratically** through the bilevel composition. To visualize this condition number amplification, we construct controlled instances of (2) with prescribed LL Hessian spectra, which isolate the analytical spectral effect with full details in Appendix A.3. Figure 1 illustrates the squared-conditioning effect as the LL Hessian becomes more spectrally imbalanced.

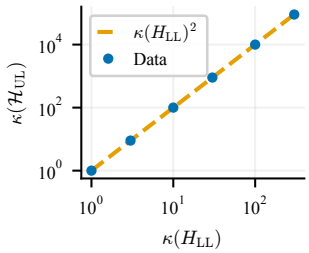


Figure 1: $\kappa(\mathcal{H}_{\mathrm{UL}}) = \kappa(H_{\mathrm{LL}})^2$.

More importantly, this amplification can appear directly in the UL hypergradient. Let h_i denote an eigenvalue of H_{LL} and x_i the corresponding component of an UL displacement aligned with the LL curvature eigendirections. The associated hypergradient singular magnitude scales as x_i/h_i^2. Appendix A.3 further illustrates this amplification and the effect of polar normalization in Figure 4.

Hence, anisotropic LL curvature can produce a UL hypergradient with highly unequal singular values, causing a FVB UL update to be dominated by a small number of spectral directions. Complete derivations are provided in Appendix A. As mentioned above, quadratic BLO provides a local second-order view of more general smooth bilevel problems, where the UL Hessian characterizes the geometry up to higher-order terms. This motivates our subsequent Muon-style orthogonalized UL updates in the next section, which regulate the singular spectrum to mitigate the spectral imbalance.

4 The Proposed BiMuon Algorithm

In this section, we first introduce the necessary background of the Muon optimizer in Section 4.1, which facilitate subsequent discussions of our BiMuon design in Section 4.2.

4.1 The Muon Optimizer: A Primer

Consider the single-level optimization $\min_{X\in\mathbb{R}^{m\times n}} \phi(X)$. Given a nonzero matrix $M \in \mathbb{R}^{m\times n}$ with compact singular value decomposition (SVD) $M = U\Sigma V^\top$, Muon (Jordan et al., 2024) constructs an orthogonalized update direction through the supported polar factor $\mathrm{polar}(M) := UV^\top$, where $\mathrm{polar}(0) := 0$. A Muon-style update takes the form

$$X_{t+1} = X_t - \beta_t\,\mathrm{polar}(M_t),$$

where M_t is the gradient or momentum matrix. The polar transformation preserves the active left and right singular subspaces of M_t, while removing its positive singular-value scaling.

However, computing the exact polar factor requires expensive SVD computations. In practical Muon implementations, polar factor is approximated by Newton-Schulz (NS) iterations based only on matrix multiplications. Following (Kim & Oh, 2026), we initialize $O^{[0]} = \frac{M}{\max\{1,\|M\|_{\mathrm{F}}\}}$, and apply

$$O^{[j+1]} = p\Big(O^{[j]}(O^{[j]})^\top\Big)\,O^{[j]}, \qquad j = 0,\ldots,q-1,$$

Algorithm 1 BiMuon: Exact-Polar and Newton–Schulz Variants

Input: Horizon T; depths D, Q, q; stepsizes α, β, η; momentum θ; initial points X_0, Y_0; batch sizes $B_{\text{low}}, B_f, B_{xy}, B_H$

Initialize $M_{-1} = 0$
for $k = 0, \dots, T-1$ **do**
  Set $Y_k^0 = Y_0$ if $k = 0$, and $Y_k^0 = Y_{k-1}^D$ otherwise
  for $t = 0, \dots, D-1$ **do**
    Draw a fresh batch $\mathcal{S}_{k,t}$ of size B_{low}; update Y_k^{t+1} using (3)
  end for
  Draw independent fresh batches $\mathcal{D}_{F,k}, \mathcal{D}_{G,k}, \{\mathcal{B}_{k,j}\}_{j=1}^{Q}$ as specified below
  Estimate $\widehat{\nabla\Phi}(X_k)$ using (4) via the Neumann recursion
  Update M_k using (5)
  Option I (Exact-Polar): $O_k = \text{polar}(M_k)$; **Option II** (Finite-NS): $O_k = \text{NS}(M_k, q)$
  $X_{k+1} = X_k - \beta O_k$
end for
return X_T

where p is a degree-d Taylor approximation of $x^{-1/2}$ and q is the NS iteration depth. We denote the output after q iterations by $\text{NS}(M, q) := O^{[q]}$. The explicit polynomial and the complete definition of $\text{NS}(M, q)$ are given in Appendix B, with the routine summarized in Algorithm 2. We note that the exact polar factor provides a useful reference transformation for theoretical understanding, while finite-NS technique gives a practical approximation using only matrix multiplications. We next incorporate these matrix-based directions into stochastic bilevel optimization.

4.2 The Proposed BiMuon Algorithm

Our BiMuon method combines the following steps: warm-started LL tracking, stochastic hypergradient estimation, and a Muon-style UL update, which are also summarized in Algorithm 1.

1) LL Trajectory Tracking. At the LL, our goal is to track the solution $Y^\star(X_k)$ of the LL problem. At outer iteration $k \geq 1$, the LL optimization is initialized using the previous LL output, $Y_k^0 = Y_{k-1}^D$, with $Y_0^0 = Y_0$. Starting from Y_k^0, we perform D stochastic gradient steps:

$$Y_k^{t+1} = Y_k^t - \alpha \nabla_Y G(X_k, Y_k^t; \mathcal{S}_{k,t}), \qquad t = 0, \dots, D-1, \tag{3}$$

where $\mathcal{S}_{k,t}$ is a fresh mini-batch of size B_{low}. The resulting Y_k^D serves as the approximate LL solution used in the UL hypergradient estimation.

The warm start is important in the bilevel setting because the target $Y^\star(X_k)$ changes as the UL variable is updated. Thus, instead of solving each LL problem from scratch, BiMuon initializes the next LL loop from the previous LL output and tracks the moving solution across outer iterations.

2) Stochastic UL Hypergradient Updates. At the UL, we update X using a stochastic hypergradient estimator based on a truncated Neumann-series approximation. At (X_k, Y_k^D), independent fresh mini-batches $\mathcal{D}_{F,k}$, $\mathcal{D}_{G,k}$, and $\{\mathcal{B}_{k,j}\}_{j=1}^{Q}$, of sizes B_f, B_{xy}, and B_H, respectively, are used to compute the partial gradients and Hessian–vector products. This yields:

$$\widehat{\nabla\Phi}(X_k) = \nabla_X F(X_k, Y_k^D; \mathcal{D}_{F,k}) - \nabla^2_{XY} G(X_k, Y_k^D; \mathcal{D}_{G,k}) \times \left[\eta \sum_{i=0}^{Q} \left\{ \prod_{j=i+1}^{Q} \left(I - \eta \nabla^2_{YY} G(X_k, Y_k^D; \mathcal{B}_{k,j}) \right) \right\} \nabla_Y F(X_k, Y_k^D; \mathcal{D}_{F,k}) \right]. \tag{4}$$

The estimator in (4) is evaluated by setting $u_Q = \nabla_Y F(X_k, Y_k^D; \mathcal{D}_{F,k})$, $u_{i-1} = (I - \eta \nabla^2_{YY} G(X_k, Y_k^D; \mathcal{B}_{k,i}))[u_i]$, and $v_Q = \eta \sum_{i=0}^{Q} u_i$, followed by $\widehat{\nabla\Phi}(X_k) = \nabla_X F(X_k, Y_k^D; \mathcal{D}_{F,k}) - \nabla^2_{XY} G(X_k, Y_k^D; \mathcal{D}_{G,k})[v_Q]$. This requires Q Hessian–vector products without forming the Hessian inverse; see Algorithm 3 in Appendix B.

Next, we aggregate this estimator using exponential moving average (EMA) momentum:

$$M_k = \theta M_{k-1} + (1-\theta)\widehat{\nabla\Phi}(X_k), \qquad M_{-1} = 0, \tag{5}$$

where $\theta \in [0, 1)$ is the momentum coefficient. UL variable X_k is updated by an orthogonalized step:

$$X_{k+1} = X_k - \beta O_k,$$

where $\beta > 0$ is the UL step-size. Algorithm 1 specifies the Exact-Polar direction in Option I and the Finite-NS direction in Option II.

5 Finite-Time Convergence Rate Analysis of BiMuon

In this section, we first establish the convergence of the Exact-Polar BiMuon version and then extend the analysis to the Finite-NS BiMuon version.

5.1 Assumptions

Toward this end, we begin with several technical assumptions that are useful in our subsequent theoretical analysis.

Assumption 1 (Strongly Convex LL Problem)**.** *The function $g(X, \cdot)$ is μ-strongly convex, with $\mu > 0$. The same property holds for $G(X, \cdot; \zeta)$ almost surely. The function Φ may be nonconvex and satisfies $\Phi_{\inf} := \inf_X \Phi(X) > -\infty$.*

Assumption 2 (Lipschitz Continuity and Smoothness)**.** *The functions f, F are continuously differentiable and g, G are twice continuously differentiable. There are constants $C_f, L > 0$ and $\tau, \rho \geq 0$ such that, for all $(X, Y), (X', Y')$ and almost every sample, $|F(X, Y; \xi) - F(X', Y'; \xi)| \leq C_f\|(X - X', Y - Y')\|_{\mathrm{F}}$, $\|\nabla F(X, Y; \xi) - \nabla F(X', Y'; \xi)\|_{\mathrm{F}} \leq L\|(X - X', Y - Y')\|_{\mathrm{F}}$, $\|\nabla G(X, Y; \zeta) - \nabla G(X', Y'; \zeta)\|_{\mathrm{F}} \leq L\|(X - X', Y - Y')\|_{\mathrm{F}}$, $\|\nabla^2_{XY} G(X, Y; \zeta) - \nabla^2_{XY} G(X', Y'; \zeta)\|_{\mathrm{op}} \leq \tau\|(X - X', Y - Y')\|_{\mathrm{F}}$, and $\|\nabla^2_{YY} G(X, Y; \zeta) - \nabla^2_{YY} G(X', Y'; \zeta)\|_{\mathrm{op}} \leq \rho\|(X - X', Y - Y')\|_{\mathrm{F}}$. The same inequalities hold for the population functions.*

Assumption 3 (Stochastic Derivatives)**.** *All stochastic derivative oracles are unbiased. Mini-batches are i.i.d. and fresh conditional on the history; distinct batches are mutually independent. Moreover, $\mathbb{E}_\zeta\|\nabla_Y G(X, Y; \zeta) - \nabla_Y g(X, Y)\|_F^2 \leq \sigma^2$.*

These assumptions imply $\|\nabla_X F\|_{\mathrm{F}}, \|\nabla_Y F\|_{\mathrm{F}} \leq C_f$, $\|\nabla^2_{XY} G\|_{\mathrm{op}} \leq L$, and $\mu I \preceq \nabla^2_{YY} G \preceq LI$. The last two inequalities are operator bounds induced by the Frobenius norms.

Throughout this section, $D \geq 1$ and $B_{\mathrm{low}}, B_f, B_{xy}, B_H$ are fixed positive integers. Let L_Φ denote the UL smoothness constant given in Lemma 3 of Appendix C. Algorithm 1 returns X_T in practice, while the analysis uses X_R with $R \sim \mathrm{Unif}\{0, \ldots, T-1\}$ independent of the algorithmic randomness. Thus, $\mathbb{E}\|\nabla\Phi(X_R)\|_* = \frac{1}{T}\sum_{k=0}^{T-1} \mathbb{E}\|\nabla\Phi(X_k)\|_*$.

5.2 Convergence Results of BiMuon

When UL movement shifts the LL target, the resulting tracking error contributes to hypergradient bias and then to EMA error. The following bound quantifies the first part by UL momentum rank.

Lemma 1 (Rank-Dependent Upper–LL Tracking)**.** *Under Assumptions 1–3, choosing $0 < \alpha \leq 1/L$, $0 < \eta < 1/L$, $0 \leq \theta < 1$, $\beta > 0$, and $D \geq 1$ in Algorithm 1, we have*

$$\frac{1}{T}\sum_{k=0}^{T-1} \mathbb{E}\|Y_k^D - Y^\star(X_k)\|_{\mathrm{F}}^2 \leq \frac{\|Y_0 - Y^\star(X_0)\|_{\mathrm{F}}^2}{\alpha\mu T} + \frac{2\alpha\sigma^2}{\mu B_{\mathrm{low}}} + \frac{L^2\beta^2}{\alpha^2\mu^4 T}\sum_{k=0}^{T-2} \mathbb{E}\,\mathrm{rank}(M_k). \tag{6}$$

For Finite-NS BiMuon, we measure the finite-NS approximation error relative to the exact polar direction. For a run of length T, suppose that

$$\|\,\mathrm{NS}(M_k, q) - \mathrm{polar}(M_k)\|_{\mathrm{op}} \leq \varepsilon_q, \qquad 0 \leq k < T, \tag{7}$$

almost surely for some deterministic $\varepsilon_q \in [0, 1)$. Here, ε_q denotes the NS approximation error and is distinct from the target stationarity accuracy ϵ.

Lemma 2 (Finite-NS Approximation; Kim & Oh, 2026)**.** *For the degree-d Taylor NS iteration in Section 4.1, suppose that the initial orthogonality residual is uniformly bounded by $\delta_{k,0} \le \delta_0 < 1$ almost surely for all $0 \le k < T$. Then* (7) *holds with $\varepsilon_q \le 1 - \sqrt{1 - \delta_0^{(d+1)^q}} \le \delta_0^{(d+1)^q}$. Hence, for any fixed run satisfying $\delta_0 < 1$, the NS approximation error decays doubly exponentially with q.*

Due to space limitation, the residual definition and additional approximation details are provided in Appendix D. We now present the convergence guarantee for Exact-Polar and Finite-NS BiMuon.

Theorem 1 (Exact-Polar BiMuon and Finite-NS BiMuon)**.** *Suppose Assumptions 1–3 hold, and fix $\beta_0 > 0$. For any integer $T \ge 2$, run Algorithm 1 with $\alpha = 1/(L\sqrt{T})$, $\eta = 1/(2L)$, $\theta = 1 - 1/\sqrt{T}$, $\beta = \beta_0/T^{3/4}$, and $Q = \max\{1, \lceil \log T/(4|\log(1-\eta\mu)|)\rceil\}$.*

(i) Exact-Polar BiMuon: *If $O_k = \mathrm{polar}(M_k)$, then $\mathbb{E}\|\nabla\Phi(X_R)\|_* \le C_{\mathrm{pol}} T^{-1/4}$.*

(ii) Finite NS BiMuon: *If $O_k = \mathrm{NS}(M_k, q)$ for $q \ge 0$ and* (7) *holds, then $\mathbb{E}\|\nabla\Phi(X_R)\|_* \le \frac{C_{\mathrm{pol}}}{1-\varepsilon_q} T^{-1/4}$. Here C_{pol} is independent of T; its explicit expression is given in* (29) *of Appendix C. The NS rate is uniform in T whenever a common $\varepsilon_q < 1$ can be chosen independently of T.*

The proofs of these two cases are proved in Appendices C and D, respectively.

Corollary 1 (Iteration and Stochastic Derivative Complexity)**.** *Under the conditions of Theorem 1, with a common $\varepsilon_q < 1$ independent of T for Finite-NS* BiMuon*, both Exact-Polar* BiMuon *and Finite-NS* BiMuon *require $T = O(\epsilon^{-4})$ outer iterations to obtain X_R with $\mathbb{E}\|\nabla\Phi(X_R)\|_* \le \epsilon$, our criterion for ϵ-stationarity. Under the oracle schedule of Theorem 1, the corresponding stochastic derivative complexity is $\widetilde{O}(\epsilon^{-4})$. Detailed derivative counts are given in Appendix D.*

Remark 1 (Finite-NS and Exact-Polar). For each fixed run, Lemma 2 shows that the NS approximation error decreases doubly exponentially as the depth q increases. Hence, the finite-NS result approaches that of the Exact-Polar rapidly with increasing q.

Remark 2. The resulting $O(T^{-1/4})$ convergence rate *matches* the best-known rate for stochastic single-level Muon-type optimization under matrix parameterizations (Kim & Oh, 2026). Thus, despite the additional LL tracking and hypergradient estimation errors introduced by the bilevel structure, BiMuon preserves the convergence order of its single-level counterparts. Here, stationarity is measured in the nuclear norm, consistent with the spectral geometry of the polar update.

6 BiMuon WITH BLOCKWISE DIAGONAL MATRICES AND RELATIVE STEP-SIZE ALLOCATION

To improve the computational and parallel efficiency of BiMuon, we consider BiMuon with blockwise diagonal matrices (called *Blockwise Matrix* BiMuon for convenience subsequently), where the Muon-style orthogonalized direction can be computed separately in parallel for each UL matrix block, followed by a block-specific relative stepsize. Note that modern neural networks contain many special structured matrix-valued parameter blocks that are *weakly or sparsely* coupled through the network. Jointly processing all matrix parameters can be expensive, whereas blockwise processing reduces matrix sizes and supports parallel computation (Zhang et al., 2017; Mendler-Dünner & Lucchi, 2020; Bahamou et al., 2023) without incurring high inaccuracy. Next, we restate a theoretical result from Amari et al. (2019) to motivate our blockwise design.

Example 1 (Motivating Example for Blockwise Processing)**.** Under the wide random-network assumptions of (Amari et al., 2019), the Fisher information matrix is approximately block diagonal, with off-block terms of order $O_p(n^{-1/2})$, where n is the hidden-layer width. Thus, in this regime, cross-block interactions in the Fisher geometry become weak as the network width increases.

It is worth noting that the blockwise diagonal matrix setting is also consistent with using Muon for LLM training: in Transformer training, applying Muon separately to the query, key, and value matrices has been reported to perform better than treating a fused QKV projection as a single matrix (Jordan et al., 2024).

We write the UL variable as $X = (X^{(1)}, \ldots, X^{(J)})$, where $X^{(b)} \in \mathbb{R}^{m_b \times n_b}$ denotes the bth matrix block. Let $M_k^{(b)}$ and $O_k^{(b)}$ denote its hypergradient momentum and Muon-style orthogonalized

direction, respectively. Blockwise Matrix BiMuon at the UL updates can be written as

$$X_{k+1}^{(b)} = X_k^{(b)} - \beta\omega_{b,k}O_k^{(b)}, \qquad \omega_{b,k} \geq 0, \tag{8}$$

where $\beta > 0$ is the global UL stepsize and $\omega_{b,k}$ determines the relative stepsize of block b. Conceptually, the block momenta may be organized as $\overline{M}_k = \mathrm{blkdiag}(M_k^{(1)}, \ldots, M_k^{(J)})$, so that the Muon directions are computed on smaller matrices in parallel.

Importantly, the blockwise implementation does not require the UL matrix blocks to be independent. Instead, we control their simultaneous effect on the moving LL target. Suppose that, for constants $L_{XY,b} > 0$, $\|\nabla_Y g(X,Y) - \nabla_Y g(X',Y)\|_{\mathrm{F}}^2 \leq \sum_{b=1}^J L_{XY,b}^2 \|X^{(b)} - X'^{(b)}\|_{\mathrm{F}}^2$. This is a joint bound on simultaneous changes of all matrix blocks rather than an independence assumption. Let $r_b := \min\{m_b, n_b\}$ and define the blockwise LL sensitivity $w_b := \frac{L_{XY,b}^2 r_b}{\mu^2}$. Since $\|O_k^{(b)}\|_{\mathrm{F}}^2 \leq r_b$, the μ-strong convexity of the LL problem yields $\|Y^\star(X_{k+1}) - Y^\star(X_k)\|_{\mathrm{F}}^2 \leq \beta^2 \sum_{b=1}^J w_b \omega_{b,k}^2$. Hence, w_b quantifies how strongly the movement of block b can affect the LL target while remaining valid under joint block updates. In particular, blocks with larger w_b should be updated more conservatively.

The LL sensitivity alone, however, should not determine the relative stepsize, since a sensitive block may also provide a strong UL signal. We therefore define $a_{b,k} := \langle M_k^{(b)}, O_k^{(b)} \rangle$ as the predicted UL progress associated with block b, and choose the relative stepsizes through

$$\max_{\omega_{1,k},\ldots,\omega_{J,k} \geq 0} \sum_{b=1}^J a_{b,k}\omega_{b,k} \qquad \text{s.t.} \qquad \sum_{b=1}^J w_b \omega_{b,k}^2 \leq 1. \tag{9}$$

For nonnegative scores that are not all zero, weighted Cauchy–Schwarz yields

$$\omega_{b,k} = \frac{a_{b,k}/w_b}{\sqrt{\sum_{c=1}^J a_{c,k}^2/w_c}}. \tag{10}$$

Thus, blocks with stronger predicted UL progress receive larger relative stepsizes, whereas blocks with greater influence on the LL target are more strongly regulated. Blockwise Matrix BiMuon therefore retains the computational and parallel advantages of separate Muon-style orthogonalization while using the bilevel structure to guide relative stepsize allocation.

7 Experimental Results

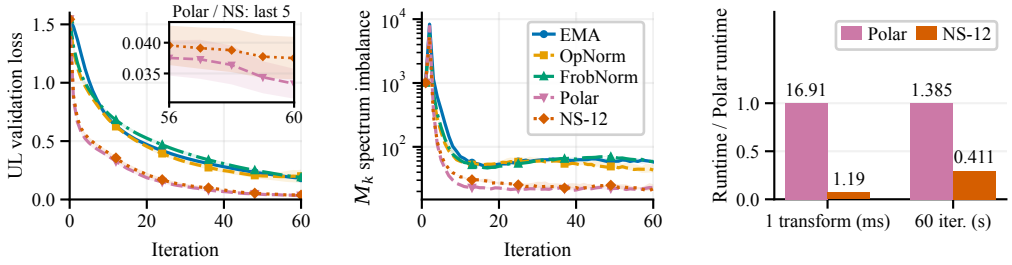
We evaluate BiMuon on two real-data tasks: CIFAR-10 representation learning and CIFAR-100 blockwise relative stepsize allocation with controlled block sensitivities.

7.1 Real-Data Representation Learning

We evaluate BiMuon on a CIFAR-10 (Krizhevsky, 2009) classification task using frozen pretrained ResNet-18 (He et al., 2016) features and a matrix-valued UL representation $X \in \mathbb{R}^{256\times512}$.

1) Setup. The LL variable is a regularized linear classifier trained on the training split, while the UL objective is cross-entropy on a separate validation split. All methods use the same stocBiO (Ji et al., 2021) LL solver, stochastic implicit hypergradient estimator, and EMA momentum defined in (5); only the final transformation of the UL momentum differs. We use finite Newton–Schulz depth $q = 12$ for NS-BiMuon. Here, optimized UL validation loss denotes validation cross-entropy after independently refitting the LL classifier at each logged X_k. Full data splits, solver parameters and tuning details are provided in Appendix F.1.

2) Baselines. We compare five UL updates. **stocBiO+EMA** directly applies the hypergradient momentum. **OpNorm** and **FrobNorm** normalize it by its operator and Frobenius norms, respectively; both change only the global update scale and preserve the relative singular spectrum. Our proposed **exact-polar** BiMuon uses the supported polar factor. Its finite-NS variant (called **NS-**BiMuon for convenience) approximates this transformation using 12 NS iterations. All methods share the same stochastic derivative budget and use stepsizes selected under the same tuning protocol.



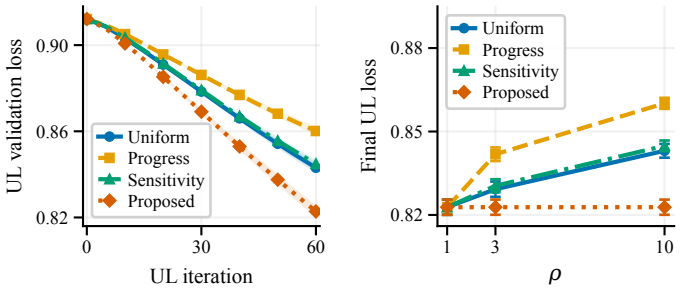
(a) UL optimization comparison (b) Momentum spectrum diagnostic (c) Polar vs. NS runtime

Figure 2: CIFAR-10 representation learning and A100 runtime comparison.

3) Results. Figure 2(a) shows that Exact-Polar BiMuon and Finite-NS BiMuon achieve substantially lower optimized UL validation loss than stocBiO+EMA and the scalar-normalized baselines, supporting spectral reshaping beyond global rescaling. Figure 2(b) shows lower spectral imbalance of the pre-transformation momentum M_k under both BiMuon variants; the metric is the P90/P10 singular-value ratio defined in Appendix F.1. Figure 2(c) shows that NS-12 is $14.28\times$ faster than Exact-Polar per UL transformation and $3.38\times$ faster over 60 updates on an A100 GPU.

7.2 Blockwise Stepsize Allocation on CIFAR-100

We further evaluate the blockwise relative-stepsize allocation in Section 6 on CIFAR-100 (Krizhevsky, 2009).



(a) $\rho = 10$ trajectory (b) Final UL loss vs. ρ

Figure 3: CIFAR-100 blockwise allocation.

1) Setup. We use frozen image features and partition the LL classifier into four matrix blocks. To obtain controlled and exactly known block sensitivities, block b is coupled to its UL matrix through a coefficient c_b, yielding $w_b = L_{XY,b}^2 r_b/\mu^2 = c_b^2 r_b$. We vary the sensitivity heterogeneity through ρ and evaluate $\rho \in \{1, 3, 10\}$. Here, $\rho = \max_b c_b / \min_b c_b$ is the sensitivity ratio, distinct from the Hessian-Lipschitz constant in Assumption 2. All compared allocations use the same bilevel estimator, blockwise update-direction construction, global stepsize, and theoretical LL-drift budget $\sum_b w_b \omega_{b,k}^2 = 1$. Full construction details are given in Appendix F.2.

2) Baselines. We compare four relative-stepsize rules: **Uniform**, which ignores block differences; **Progress-only**, based on the UL score $a_{b,k}$; **Sensitivity-only**, based on $1/w_b$; and the **Proposed** allocation, based on $a_{b,k}/w_b$. Thus, the comparison isolates whether UL progress and LL sensitivity should be considered jointly when distributing the same blockwise movement budget.

3) Results. Figure 3 shows that the proposed allocation achieves the best UL optimization among the tested rules. For both $\rho = 3$ and $\rho = 10$, it outperforms Uniform, Progress-only, and Sensitivity-only on all five evaluation seeds. The advantage becomes clearer when block sensitivities are more heterogeneous, supporting the use of both $a_{b,k}$ and w_b in the relative-stepsize rule.

8 Conclusion

In this paper, we proposed a Muon-type approach called BiMuon for bilevel optimization and established $O(T^{-1/4})$ expected nuclear-norm stationarity convergence and $\widetilde{O}(\epsilon^{-4})$ stochastic derivative complexity guarantees for its exact-polar and finite-NS variants. For increased computation efficiency, we further investigated BiMuon with blockwise diagonal approximation and derived a bilevel-informed relative step-size allocation rule that balances UL progress and LL sensitivity across matrix blocks. Experiments on representation learning and blockwise bilevel optimization problems verified the efficacy of our proposed BiMuon designs.

AI USE DISCLOSURE

In this work, we did not use generative AI tools for any tasks requiring disclosure. Additionally, we used generative AI tools to create or modify scientific figures or images, source or search for information, edit a research paper to improve readability, and format references. We have reviewed all AI-assisted work. Literature claims and citations were checked against the original sources, and all AI-assisted text and figures were reviewed by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Complete proofs and algorithmic details are provided in the appendices. Experimental settings, data processing, hyperparameters, tuning protocols, and implementation details are reported in the appendices material to support reproducibility.

REFERENCES

Shun-ichi Amari, Ryo Karakida, and Masafumi Oizumi. Fisher information and natural gradient learning in random deep networks. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 694–702. PMLR, 2019. URL https://proceedings.mlr.press/v89/amari19a.html.

Achraf Bahamou, Donald Goldfarb, and Yi Ren. A mini-block Fisher method for deep neural networks. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 9191–9220. PMLR, 2023. URL https://proceedings.mlr.press/v206/bahamou23a.html.

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

Souradip Chakraborty, Amrit Singh Bedi, Alec Koppel, Dinesh Manocha, Huazheng Wang, Mengdi Wang, and Furong Huang. PARL: A unified framework for policy alignment in reinforcement learning from human feedback, 2023. URL https://arxiv.org/abs/2308.02585. arXiv:2308.02585.

Tianyi Chen, Yuejiao Sun, Quan Xiao, and Wotao Yin. A single-timescale method for stochastic bilevel optimization. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 2466–2488. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/chen22e.html.

Tianshu Chu, Dachuan Xu, Wei Yao, and Jin Zhang. SPABA: A single-loop and probabilistic stochastic bilevel algorithm achieving optimal sample complexity. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8848–8903. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/chu24a.html.

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 26698–26710. Curran Associates, Inc., 2022. doi: 10.52202/068431-1936. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/aa84ec1ac3f5fdcf77bce2c22705ab77-Paper-Conference.pdf.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. URL https://www.image-net.org/static_files/papers/imagenet_cvpr09.pdf.

Mucong Ding, Souradip Chakraborty, Vibhu Agrawal, Zora Che, Alec Koppel, Mengdi Wang, Amrit Bedi, and Furong Huang. SAIL: Self-improving efficient online alignment of large language models, 2024. URL https://arxiv.org/abs/2406.15567. arXiv:2406.15567.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1568–1577. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/franceschi18a.html.

Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming, 2018. URL https://arxiv.org/abs/1802.02246. arXiv:1802.02246.

Riccardo Grazzi, Massimiliano Pontil, and Saverio Salzo. Convergence properties of stochastic hypergradients. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pp. 3826–3834. PMLR, 2021. URL https://proceedings.mlr.press/v130/grazzi21a.html.

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

Andi Han, Bamdev Mishra, Pratik Jawanpuria, and Akiko Takeda. A framework for bilevel optimization on Riemannian manifolds. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.neurips.cc/paper_files/paper/2024/hash/bc1c5e5fb8ed1ef9b9b5abced2022e40-Abstract-Conference.html.

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

Feihu Huang, Junyi Li, Shangqian Gao, and Heng Huang. Enhanced bilevel optimization via bregman distance. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b9e98316cb72fee82cc1160da5810abc-Abstract-Conference.html.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 4882–4892. PMLR, 2021. URL https://proceedings.mlr.press/v139/ji21c.html.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/. Author's technical blog, December 8.

Feiyang Kang, Yifan Sun, Bingbing Wen, Si Chen, Dawn Song, Rafid Mahmood, and Ruoxi Jia. AutoScale: Scale-aware data mixing for pre-training LLMs. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=rujwIvjooA.

Gyu Yeol Kim and Min-hwan Oh. Convergence of Muon with Newton–Schulz. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/4c0129f1f7a036af5ed2409f1f9f8131-Abstract-Conference.html.

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

Jiaxiang Li and Shiqian Ma. Riemannian bilevel optimization. *Journal of Machine Learning Research*, 26(18):1–44, 2025. URL https://www.jmlr.org/papers/v26/24-0397.html.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989. doi: 10.1007/BF01589116. URL https://link.springer.com/article/10.1007/BF01589116.

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

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 1540–1552. PMLR, 2020. URL https://proceedings.mlr.press/v108/lorraine20a.html.

Celestine Mendler-Dünner and Aurelien Lucchi. Randomized block-diagonal preconditioning for parallel learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6841–6851. PMLR, 2020. URL https://proceedings.mlr.press/v119/mendler-dunner20a.html.

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

Wei Shen, Ruichuan Huang, Minhui Huang, Cong Shen, and Jiawei Zhang. On the convergence analysis of Muon, 2025. URL https://arxiv.org/abs/2505.23737.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. SOAP: Improving and stabilizing Shampoo using Adam for language modeling. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e988664070e9591f93fdcf605f7dc623-Abstract-Conference.html.

Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/71cc107d2e0408e60a3d3c44f47507bd-Abstract.html.

Huishuai Zhang, Caiming Xiong, James Bradbury, and Richard Socher. Block-diagonal Hessian-free optimization for training neural networks, 2017. URL https://arxiv.org/abs/1712.07296.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61121–61143. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhao24s.html.

A Derivations and Details for Bilevel-Induced Spectral Imbalance

This appendix gives the derivations supporting Section 3.2. We first derive the LL solution and induced UL curvature, then make the aligned hypergradient calculation explicit and describe the controlled numerical construction used in Figures 1 and 4.

A.1 LL solution and induced UL curvature

For the model in Section 3.2, let $Z_{\mathrm{tr}} \in \mathbb{R}^{n_{\mathrm{tr}} \times m}$, $Z_{\mathrm{val}} \in \mathbb{R}^{n_{\mathrm{val}} \times m}$, $R_{\mathrm{tr}} \in \mathbb{R}^{n_{\mathrm{tr}} \times n}$, and $R_{\mathrm{val}} \in \mathbb{R}^{n_{\mathrm{val}} \times n}$, with positive sample counts. The matrices X, Y, X_{prior} lie in $\mathbb{R}^{m \times n}$, and $\mu > 0$, $\gamma \geq 0$. Using the notation of the main text, let $H_{\mathrm{LL}} = \frac{Z_{\mathrm{tr}}^\top Z_{\mathrm{tr}}}{n_{\mathrm{tr}}} + \mu I$, $\Sigma_{\mathrm{val}} = \frac{Z_{\mathrm{val}}^\top Z_{\mathrm{val}}}{n_{\mathrm{val}}}$, and $H_{\mathrm{LL}} = P \operatorname{diag}(h_1, \ldots, h_m) P^\top$, where P is orthogonal and $h_i \geq \mu > 0$.

LL solution and sensitivity. Direct differentiation gives $\nabla_Y g(X, Y) = H_{\mathrm{LL}} Y - \frac{Z_{\mathrm{tr}}^\top R_{\mathrm{tr}}}{n_{\mathrm{tr}}} - \mu X$, and $\nabla^2_{YY} g(X, Y)[\Delta Y] = H_{\mathrm{LL}} \Delta Y$. Since $H_{\mathrm{LL}} \succeq \mu I \succ 0$, the unique LL solution is

$$Y^\star(X) = H_{\mathrm{LL}}^{-1} \left(\frac{Z_{\mathrm{tr}}^\top R_{\mathrm{tr}}}{n_{\mathrm{tr}}} + \mu X \right). \tag{11}$$

Differentiating this affine map gives $DY^\star(X)[\Delta X] = \mu H_{\mathrm{LL}}^{-1} \Delta X$. The inverse acts on the left; the perturbations and solution have shape $m \times n$.

UL derivatives. The partial derivatives are $\nabla_X f = \gamma(X - X_{\mathrm{prior}})$ and $\nabla_Y f = \Sigma_{\mathrm{val}} Y - Z_{\mathrm{val}}^\top R_{\mathrm{val}} / n_{\mathrm{val}}$. The chain rule and symmetry of H_{LL}^{-1} yield

$$\nabla \Phi(X) = \gamma(X - X_{\mathrm{prior}}) + \mu H_{\mathrm{LL}}^{-1} \left(\Sigma_{\mathrm{val}} Y^\star(X) - \frac{Z_{\mathrm{val}}^\top R_{\mathrm{val}}}{n_{\mathrm{val}}} \right). \tag{12}$$

Differentiating once more gives $\mathcal{H}_{\mathrm{UL}}[\Delta X] = \nabla^2 \Phi(X)[\Delta X] = \left(\mu^2 H_{\mathrm{LL}}^{-1} \Sigma_{\mathrm{val}} H_{\mathrm{LL}}^{-1} + \gamma I \right) \Delta X$. The operator acts identically and independently on each column. The target terms and prior affect the gradient and optimum but not this constant Hessian. For general $\Sigma_{\mathrm{val}} \succeq 0$, the Hessian is positive semidefinite; it is positive definite if $\Sigma_{\mathrm{val}} \succ 0$ or $\gamma > 0$. No finite condition number is claimed in a singular case.

Squared conditioning under isotropic validation. When $\Sigma_{\mathrm{val}} = I$, the matrices $P e_i e_j^\top$, $1 \leq i \leq m$, $1 \leq j \leq n$, form an orthonormal Frobenius-space eigenbasis of $\mathcal{H}_{\mathrm{UL}}$ with eigenvalues $\mu^2 / h_i^2 + \gamma$. Here e_i, e_j are the standard coordinate vectors in the respective spaces. Writing $h_{\min} = \min_i h_i$ and $h_{\max} = \max_i h_i$ gives $\kappa(\mathcal{H}_{\mathrm{UL}}) = \frac{\gamma + \mu^2 / h_{\min}^2}{\gamma + \mu^2 / h_{\max}^2}$. In particular, when $\gamma = 0$, $\mathcal{H}_{\mathrm{UL}}[\Delta X] = \mu^2 H_{\mathrm{LL}}^{-2} \Delta X$, $\kappa(\mathcal{H}_{\mathrm{UL}}) = \left(\frac{h_{\max}}{h_{\min}} \right)^2 = \kappa(H_{\mathrm{LL}})^2$. This is the identity stated in Section 3.2. The common factor μ^2 cancels from the condition number, whereas positive γ moves the ratio toward one. More generally, if $(1 - \delta) c I \preceq \Sigma_{\mathrm{val}} \preceq (1 + \delta) c I$ for $c > 0$ and $0 \leq \delta < 1$, congruence gives $(1 - \delta) c H_{\mathrm{LL}}^{-2} \preceq H_{\mathrm{LL}}^{-1} \Sigma_{\mathrm{val}} H_{\mathrm{LL}}^{-1} \preceq (1 + \delta) c H_{\mathrm{LL}}^{-2}$. Thus approximate isotropy gives spectral bounds, not exact equality or exact polar invariance.

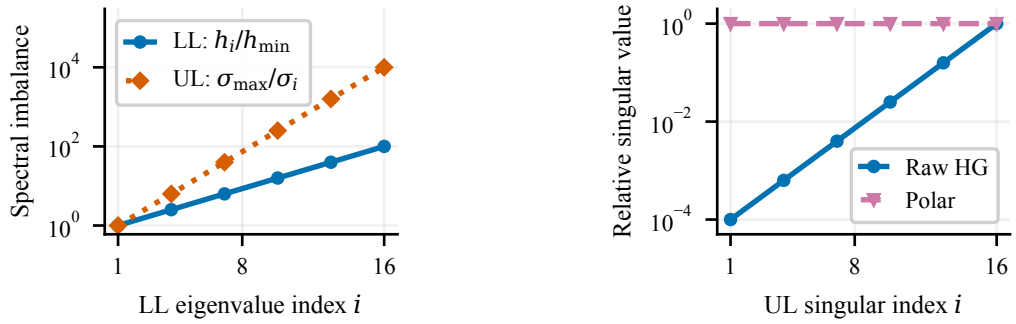
A.2 Aligned Hypergradient Singular Magnitudes and Scope

Under $\Sigma_{\mathrm{val}} = I$ and $\gamma = 0$, define $X^\star = \mu^{-1} \left(H_{\mathrm{LL}} \frac{Z_{\mathrm{val}}^\top R_{\mathrm{val}}}{n_{\mathrm{val}}} - \frac{Z_{\mathrm{tr}}^\top R_{\mathrm{tr}}}{n_{\mathrm{tr}}} \right)$. Substituting (11) into the UL gradient gives $\nabla \Phi(X) = \mu^2 H_{\mathrm{LL}}^{-2} (X - X^\star)$. Let $r = \min\{m, n\}$, let P_r contain r eigenvectors of H_{LL}, and let $V \in \mathbb{R}^{n \times r}$ satisfy $V^\top V = I_r$. For an aligned displacement $X - X^\star = P_r \operatorname{diag}(x_1, \ldots, x_r) V^\top$, $x_i > 0$, we obtain $\nabla \Phi(X) = \mu^2 P_r \operatorname{diag} \left(\frac{x_1}{h_1^2}, \ldots, \frac{x_r}{h_r^2} \right) V^\top$. Hence, the hypergradient singular magnitudes scale as $\mu^2 x_i / h_i^2$, showing explicitly how LL curvature can induce spectral imbalance in the UL hypergradient. Moreover, $\operatorname{polar}(\nabla \Phi(X)) = P_r V^\top$,

so the exact polar direction removes the positive singular-value scaling while preserving the active singular subspaces.

These identities rely on the quadratic model, isotropic validation curvature, $\gamma = 0$, and the aligned displacement above. The example therefore provides a local second-order mechanism for the spectral effect discussed in Section 3.2, rather than a general identity for arbitrary smooth bilevel problems.

A.3 Controlled Numerical Illustration of the Section 3.2 Quadratic Model



(a) Spectral amplification

(b) Polar normalization

Figure 4: Additional illustrations of the quadratic mechanism in (2). (a) For aligned $x_i = 1$, a 10^2 LL Hessian spectral spread induces a 10^4 UL hypergradient singular-value spread. (b) The supported polar factor of the same hypergradient has unit singular values; raw singular values are normalized by their maximum.

We use the quadratic model in Section 3.2 with $m = n = n_{\mathrm{tr}} = n_{\mathrm{val}} = 16$, $\mu = 0.1$, $\gamma = 0$, and $R_{\mathrm{tr}} = R_{\mathrm{val}} = X_{\mathrm{prior}} = 0$. We set $Z_{\mathrm{val}} = \sqrt{n_{\mathrm{val}}} I$, so that $Z_{\mathrm{val}}^\top Z_{\mathrm{val}}/n_{\mathrm{val}} = I$.

For each prescribed condition number $\kappa \in \{1, 3, 10, 30, 100, 300\}$, we generate an orthogonal matrix P and choose geometrically spaced eigenvalues $h_i \in [1, \kappa]$. We then define $H_{\mathrm{LL}} = P\,\mathrm{diag}(h_1, \ldots, h_m)P^\top$, $Z_{\mathrm{tr}} = \sqrt{n_{\mathrm{tr}}}\ \mathrm{diag}\big(\sqrt{h_i - \mu}\big)P^\top$, which gives $Z_{\mathrm{tr}}^\top Z_{\mathrm{tr}}/n_{\mathrm{tr}} + \mu I = H_{\mathrm{LL}}$.

Since the targets are zero, $X^\star = 0$. We use the aligned displacement $X - X^\star = P\,\mathrm{diag}(x_1, \ldots, x_m)V^\top$, with $x_i = 1$ and orthogonal V. Figure 1 reports the resulting relation $\kappa(H_{\mathrm{UL}}) = \kappa(H_{\mathrm{LL}})^2$. Figure 4 uses the $\kappa = 100$ instance with seed 0 to visualize the corresponding hypergradient spectral amplification and exact-polar normalization.

For numerical verification, we repeat each condition number over five independent orthogonal rotations using seeds 0–4. All computations are performed in float64 on CPU.

B Algorithm Implementation Details

This appendix gives the finite Newton–Schulz (NS) map and stochastic Neumann recursion used in Section 4.

B.1 Newton–Schulz Orthogonalization

Following Kim & Oh (2026), for an integer $d \geq 1$, define the degree-d Taylor polynomial of $x^{-1/2}$ around $x = 1$ as $p(x) = \sum_{\ell=0}^{d} \frac{(2\ell)!}{4^\ell(\ell!)^2}(1-x)^\ell$. For a square matrix A, its matrix evaluation is $p(A) = \sum_{\ell=0}^{d} \frac{(2\ell)!}{4^\ell(\ell!)^2}(I-A)^\ell$. Given $M \in \mathbb{R}^{m\times n}$, Algorithm 2 defines the finite-NS map $\mathrm{NS}(M, q)$ used in Section 4.1.

By construction, $\mathrm{NS}(M, 0) = O^{[0]}$ and $\mathrm{NS}(0, q) = 0$.

Algorithm 2 Newton–Schulz orthogonalization

Input: Matrix $M \in \mathbb{R}^{m\times n}$; integer NS depth $q \geq 0$; degree-d polynomial p

$O^{[0]} \leftarrow M/\max\{1, \|M\|_{\mathrm{F}}\}$
for $j = 0, \ldots, q-1$ **do**
  $O^{[j+1]} \leftarrow p\big(O^{[j]}(O^{[j]})^\top\big)\, O^{[j]}$
end for
return $\mathrm{NS}(M, q) = O^{[q]}$

B.2 Stochastic Neumann Hypergradient Estimation

The truncated Neumann term in (4) is evaluated through the Hessian–vector recursion in Algorithm 3. For fixed (X_k, Y_k^D), it uses Q independent Hessian mini-batches and requires Q Hessian–vector products without explicitly forming the LL Hessian inverse.

Algorithm 3 Stochastic inverse-Hessian estimate

Input: Fixed (X_k, Y_k^D); stepsize η; depth Q; loss batch $\mathcal{D}_{F,k}$; independent Hessian batches $\{\mathcal{B}_{k,i}\}_{i=1}^{Q}$, each of size B_H

$u_Q \leftarrow \nabla_Y F(X_k, Y_k^D; \mathcal{D}_{F,k})$
for $i = Q, \ldots, 1$ **do**
  $u_{i-1} \leftarrow u_i - \eta \nabla^2_{YY} G(X_k, Y_k^D; \mathcal{B}_{k,i})[u_i]$
end for
return $v_Q = \eta \sum_{i=0}^{Q} u_i$

Using an independent mixed-derivative batch $\mathcal{D}_{G,k}$, the resulting stochastic hypergradient estimator is $\widehat{\nabla}\Phi(X_k) = \nabla_X F(X_k, Y_k^D; \mathcal{D}_{F,k}) - \nabla^2_{XY} G(X_k, Y_k^D; \mathcal{D}_{G,k})[v_Q]$, which is the estimator in (4).

C Convergence proof for exact-polar BiMuon

This appendix proves the exact-polar result in Theorem 1; the same bilevel tracking bounds are reused for the finite-NS analysis in Appendix D. Throughout this section, suppose Assumptions 1–3 hold. The Neumann estimator is specified in Appendix B.

Lemma 3 (Smoothness of the UL objective)**.** *Under Assumptions 1–3, the UL objective Φ is L_Φ-smooth: for all $X, X' \in \mathbb{R}^{m\times n}$,*

$$\|\nabla\Phi(X) - \nabla\Phi(X')\|_{\mathrm{F}} \leq L_\Phi \|X - X'\|_{\mathrm{F}}, \tag{13}$$

where $L_\Phi = \Big(L + \frac{L^2}{\mu} + \frac{C_f\tau}{\mu} + \frac{LC_f\rho}{\mu^2}\Big)\Big(1 + \frac{L}{\mu}\Big)$.

Proof. For fixed X, strong convexity gives

$$g(X,Y) \geq g(X,0) + \langle \nabla_Y g(X,0), Y\rangle + \frac{\mu}{2}\|Y\|_{\mathrm{F}}^2 \geq g(X,0) - \|\nabla_Y g(X,0)\|_{\mathrm{F}}\|Y\|_{\mathrm{F}} + \frac{\mu}{2}\|Y\|_{\mathrm{F}}^2.$$

The right side tends to infinity as $\|Y\|_{\mathrm{F}} \to \infty$. Continuity therefore gives existence of a minimizer, and strong convexity gives uniqueness. The first-order condition is $\nabla_Y g(X, Y^\star(X)) = 0$. Strong monotonicity at fixed X and the first-order conditions at X, X' imply

$$\begin{aligned}\mu\|Y^\star(X) - Y^\star(X')\|_{\mathrm{F}}^2 &\leq \langle \nabla_Y g(X, Y^\star(X)) - \nabla_Y g(X, Y^\star(X')), Y^\star(X) - Y^\star(X')\rangle \\ &= \langle \nabla_Y g(X', Y^\star(X')) - \nabla_Y g(X, Y^\star(X')), Y^\star(X) - Y^\star(X')\rangle \\ &\leq L\|X - X'\|_{\mathrm{F}}\|Y^\star(X) - Y^\star(X')\|_{\mathrm{F}}.\end{aligned}$$

If the LL solutions differ, divide by their distance; otherwise the following bound is immediate:

$$\|Y^\star(X) - Y^\star(X')\|_{\mathrm{F}} \leq \frac{L}{\mu}\|X - X'\|_{\mathrm{F}}. \tag{14}$$

The map $(X, Y) \mapsto \nabla_Y g(X, Y)$ is continuously differentiable, with invertible derivative $\nabla^2_{YY} g(X, Y) \succeq \mu I$ in its LL argument. The implicit function theorem gives a continuously differentiable local solution near every X. Uniqueness makes the local solutions agree, so $Y^\star$ is continuously differentiable globally. Differentiating the LL first-order condition in an arbitrary UL direction U gives

$$\begin{aligned} 0 &= D_X(\nabla_Y g)(X, Y^\star(X))[U] + \nabla^2_{YY} g(X, Y^\star(X))[DY^\star(X)[U]], \\ DY^\star(X)[U] &= -(\nabla^2_{YY} g(X, Y^\star(X)))^{-1} D_X(\nabla_Y g)(X, Y^\star(X))[U]. \end{aligned}$$

The chain rule gives

$$\begin{aligned} D\Phi(X)[U] &= \langle \nabla_X f(X, Y^\star(X)), U\rangle + \langle \nabla_Y f(X, Y^\star(X)), DY^\star(X)[U]\rangle \\ &= \langle \nabla_X f(X, Y^\star(X)), U\rangle \\ &\quad - \Big\langle \nabla^2_{XY} g(X, Y^\star(X))[(\nabla^2_{YY} g(X, Y^\star(X)))^{-1} \nabla_Y f(X, Y^\star(X))], U \Big\rangle. \end{aligned}$$

Here the LL Hessian is self-adjoint, and the mixed derivative is the adjoint of $D_X(\nabla_Y g)$. This is the matrix form of the implicit hypergradient in (1). The bounds on the two loss partials, mixed derivative and inverse Hessian also give $\|\nabla\Phi(X)\|_{\mathrm{F}} \le C_f + LC_f/\mu$.

For two pairs $(X, Y), (X', Y')$, the inverse identity gives

$$\begin{aligned} &(\nabla^2_{YY} g(X, Y))^{-1} - (\nabla^2_{YY} g(X', Y'))^{-1} \\ &= (\nabla^2_{YY} g(X, Y))^{-1} [\nabla^2_{YY} g(X', Y') - \nabla^2_{YY} g(X, Y)](\nabla^2_{YY} g(X', Y'))^{-1}, \\ &\|(\nabla^2_{YY} g(X, Y))^{-1} - (\nabla^2_{YY} g(X', Y'))^{-1}\|_{\mathrm{op}} \le \frac{\rho}{\mu^2} \|(X - X', Y - Y')\|_{\mathrm{F}}. \end{aligned}$$

To bound the difference of the three-factor term $\nabla^2_{XY} g[(\nabla^2_{YY} g)^{-1} \nabla_Y f]$, change the mixed derivative first, then the inverse Hessian, and finally the loss partial gradient. Their respective norm bounds are $\frac{C_f \tau}{\mu} \|(X - X', Y - Y')\|_{\mathrm{F}}$, $\frac{LC_f \rho}{\mu^2} \|(X - X', Y - Y')\|_{\mathrm{F}}$, $\frac{L^2}{\mu} \|(X - X', Y - Y')\|_{\mathrm{F}}$. Adding the L-Lipschitz UL partial gradient proves a joint Lipschitz bound with coefficient $L + \frac{L^2}{\mu} + \frac{C_f \tau}{\mu} + \frac{LC_f \rho}{\mu^2} \le L_\Phi$. In particular, setting $X' = X$ and $Y' = Y^\star(X)$ yields the estimate used below to control hypergradient bias:

$$\begin{aligned} &\|\nabla_X f(X, Y) - \nabla^2_{XY} g(X, Y)[(\nabla^2_{YY} g(X, Y))^{-1} \nabla_Y f(X, Y)] - \nabla\Phi(X)\|_{\mathrm{F}} \\ &\quad \le L_\Phi \|Y - Y^\star(X)\|_{\mathrm{F}}. \end{aligned} \tag{15}$$

Setting $Y = Y^\star(X)$ and $Y' = Y^\star(X')$, and using (14), gives

$$\begin{aligned} \|\nabla\Phi(X) - \nabla\Phi(X')\|_{\mathrm{F}} &\le \left(L + \frac{L^2}{\mu} + \frac{C_f \tau}{\mu} + \frac{LC_f \rho}{\mu^2}\right) \left(\|X - X'\|_{\mathrm{F}} + \|Y^\star(X) - Y^\star(X')\|_{\mathrm{F}}\right) \\ &\le \left(L + \frac{L^2}{\mu} + \frac{C_f \tau}{\mu} + \frac{LC_f \rho}{\mu^2}\right) (1 + L/\mu) \|X - X'\|_{\mathrm{F}} \\ &= L_\Phi \|X - X'\|_{\mathrm{F}}. \end{aligned}$$

Finally, the fundamental theorem of calculus and (13) imply

$$\begin{aligned} \Phi(X + U) - \Phi(X) - \langle \nabla\Phi(X), U\rangle &= \int_0^1 \langle \nabla\Phi(X + tU) - \nabla\Phi(X), U\rangle \, dt \\ &\le \int_0^1 L_\Phi t \|U\|^2_{\mathrm{F}} \, dt = \frac{L_\Phi}{2} \|U\|^2_{\mathrm{F}}. \end{aligned}$$

Thus the smoothness conclusion also gives the descent inequality used in both convergence proofs:

$$\Phi(X + U) \le \Phi(X) + \langle \nabla\Phi(X), U\rangle + \frac{L_\Phi}{2} \|U\|^2_{\mathrm{F}}. \tag{16}$$

□

The following proposition bounds the truncation bias of the stochastic Neumann estimate.

Proposition 1 (Bias of the inverse-Hessian estimate)**.** *Fix (X, Y) and let v_Q be the estimate returned by Algorithm 3, evaluated at (X, Y) in place of (X_k, Y_k^D). Omit (X, Y) from the derivative arguments. Let $0 < \eta < 1/L$ and $|\mathcal{B}_i| = B_H$. Then*

$$\|\mathbb{E}v_Q - (\nabla^2_{YY} g)^{-1}\nabla_Y f\|_{\mathrm{F}} \le \frac{C_f}{\mu}(1-\eta\mu)^{Q+1}. \tag{17}$$

Proof. Algorithm 3 and $\mu I \preceq \nabla^2_{YY} G \preceq LI$ give $0 \prec I - \eta\nabla^2_{YY} G(\mathcal{B}_i) \preceq (1-\eta\mu)I$, $\|u_{i-1}\|_{\mathrm{F}} \le (1-\eta\mu)\|u_i\|_{\mathrm{F}}$, $\|u_i\|_{\mathrm{F}} \le C_f(1-\eta\mu)^{Q-i}$. Summing these bounds gives the estimate used later for the stochastic hypergradient:

$$\|v_Q\|_{\mathrm{F}} \le \eta C_f \sum_{i=0}^{Q}(1-\eta\mu)^i = \frac{C_f}{\mu}[1-(1-\eta\mu)^{Q+1}] \le \frac{C_f}{\mu} \quad \text{almost surely.} \tag{18}$$

Since u_i depends only on $u_Q, \mathcal{B}_Q, \dots, \mathcal{B}_{i+1}$,

$$\begin{aligned}
&\mathbb{E}[u_{i-1} \mid u_Q, \mathcal{B}_Q, \dots, \mathcal{B}_{i+1}] = (I - \eta\nabla^2_{YY} g)u_i,\\
&\mathbb{E}u_i = (I - \eta\nabla^2_{YY} g)^{Q-i}\nabla_Y f,\\
&\mathbb{E}v_Q = \eta\sum_{i=0}^{Q}(I - \eta\nabla^2_{YY} g)^i\nabla_Y f\\
&= [I - (I - \eta\nabla^2_{YY} g)^{Q+1}](\nabla^2_{YY} g)^{-1}\nabla_Y f,\\
&\mathbb{E}v_Q - (\nabla^2_{YY} g)^{-1}\nabla_Y f = -(I - \eta\nabla^2_{YY} g)^{Q+1}(\nabla^2_{YY} g)^{-1}\nabla_Y f,\\
&\|\mathbb{E}v_Q - (\nabla^2_{YY} g)^{-1}\nabla_Y f\|_{\mathrm{F}} \le (1-\eta\mu)^{Q+1}\mu^{-1}C_f.
\end{aligned}$$

□

We next use the bias decomposition of Lemma 7 in Ji et al. (2021). We retain the existing L_Φ as an upper bound on its local Lipschitz coefficient and the $Q+1$ truncation exponent from Proposition 1.

Lemma 4 (Conditional hypergradient bias)**.** *Let $0 < \eta < 1/L$. Conditioning on X_k and Y_k^D, we have*

$$\|\mathbb{E}\widehat{\nabla\Phi}(X_k) - \nabla\Phi(X_k)\|^2_{\mathrm{F}} \le 2L_\Phi^2\|Y_k^D - Y^\star(X_k)\|^2_{\mathrm{F}} + \frac{2L^2{C_f}^2}{\mu^2}(1-\eta\mu)^{2Q+2}. \tag{19}$$

Proof. Throughout this proof, X_k and Y_k^D are fixed, and $\mathbb{E}$ is over the fresh batches $\mathcal{D}_{\mathrm{F}}, \mathcal{D}_G, \mathcal{B}_1, \dots, \mathcal{B}_Q$. The outer index k is suppressed on these batches. All derivatives are evaluated at (X_k, Y_k^D). By the conditional independence in Assumption 3,

$$\begin{aligned}
&\mathbb{E}\widehat{\nabla\Phi}(X_k) = \nabla_X f - \nabla^2_{XY} g[\mathbb{E}v_Q],\\
&\mathbb{E}\widehat{\nabla\Phi}(X_k) - \nabla\Phi(X_k) = \nabla_X f - \nabla^2_{XY} g[(\nabla^2_{YY} g)^{-1}\nabla_Y f] - \nabla\Phi(X_k)\\
&\quad - \nabla^2_{XY} g[\mathbb{E}v_Q - (\nabla^2_{YY} g)^{-1}\nabla_Y f].
\end{aligned}$$

By (15) and (17),

$$\begin{aligned}
&\|\mathbb{E}\widehat{\nabla\Phi}(X_k) - \nabla\Phi(X_k)\|^2_{\mathrm{F}}\\
&\le 2\|\nabla_X f - \nabla^2_{XY} g[(\nabla^2_{YY} g)^{-1}\nabla_Y f] - \nabla\Phi(X_k)\|^2_{\mathrm{F}}\\
&+ 2L^2\|\mathbb{E}v_Q - (\nabla^2_{YY} g)^{-1}\nabla_Y f\|^2_{\mathrm{F}}\\
&\le 2L_\Phi^2\|Y_k^D - Y^\star(X_k)\|^2_{\mathrm{F}} + \frac{2L^2{C_f}^2}{\mu^2}(1-\eta\mu)^{2Q+2}.
\end{aligned}$$

This proves (19). □

Lemma 5 (Rank-dependent LL-solution tracking)**.** *For Algorithm 1 with either exact polar or finite NS, $0 < \alpha \le 1/L$, $\beta > 0$, and $D \ge 1$,*

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|Y_k^D - Y^\star(X_k)\|^2_{\mathrm{F}} \le \frac{\|Y_0 - Y^\star(X_0)\|^2_{\mathrm{F}}}{\alpha\mu T} + \frac{2\alpha\sigma^2}{\mu B_{\mathrm{low}}} + \frac{L^2\beta^2 r}{\alpha^2\mu^4}. \tag{20}$$

Here $r = \min\{m, n\}$ bounds the rank of the UL momentum matrix.

Proof. The fundamental theorem of calculus along the LL segment gives

$$\nabla_Y g(X_k, Y_k^t) - \nabla_Y g(X_k, Y^\star(X_k)) = \left[\int_0^1 \nabla^2_{YY} g\big(X_k, Y^\star(X_k) + u(Y_k^t - Y^\star(X_k))\big)\, du\right][Y_k^t - Y^\star(X_k)].$$

The self-adjoint operator in brackets lies between μI and LI. Since $0 < \alpha \le 1/L$, subtracting α times that operator from I gives eigenvalues in $[0, 1-\alpha\mu]$. Using the zero LL gradient at $Y^\star(X_k)$ therefore gives $\|Y_k^t - \alpha\nabla_Y g(X_k, Y_k^t) - Y^\star(X_k)\|_{\mathrm{F}}^2 \le (1-\alpha\mu)^2\|Y_k^t - Y^\star(X_k)\|_{\mathrm{F}}^2$. By freshness and unbiasedness of the LL batch,

$$\mathbb{E}[\nabla_Y G(X_k, Y_k^t; \mathcal{S}_{k,t}) - \nabla_Y g(X_k, Y_k^t) \mid X_k, Y_k^t] = 0,$$
$$\mathbb{E}[\|\nabla_Y G(X_k, Y_k^t; \mathcal{S}_{k,t}) - \nabla_Y g(X_k, Y_k^t)\|_{\mathrm{F}}^2 \mid X_k, Y_k^t] \le \sigma^2/B_{\mathrm{low}}.$$

The conditional cross term vanishes, and hence

$$\begin{aligned}
&\mathbb{E}[\|Y_k^{t+1} - Y^\star(X_k)\|_{\mathrm{F}}^2 \mid X_k, Y_k^t] \\
&= \|Y_k^t - \alpha\nabla_Y g(X_k, Y_k^t) - Y^\star(X_k)\|_{\mathrm{F}}^2 \\
&\quad + \alpha^2\mathbb{E}[\|\nabla_Y G(X_k, Y_k^t; \mathcal{S}_{k,t}) - \nabla_Y g(X_k, Y_k^t)\|_{\mathrm{F}}^2 \mid X_k, Y_k^t] \\
&\le (1-\alpha\mu)^2\|Y_k^t - Y^\star(X_k)\|_{\mathrm{F}}^2 + \frac{\alpha^2\sigma^2}{B_{\mathrm{low}}}.
\end{aligned}$$

Iterating and taking expectations,

$$\begin{aligned}
&\mathbb{E}\|Y_k^D - Y^\star(X_k)\|_{\mathrm{F}}^2 \le (1-\alpha\mu)^{2D}\mathbb{E}\|Y_k^0 - Y^\star(X_k)\|_{\mathrm{F}}^2 \\
&+ \frac{\alpha^2\sigma^2}{B_{\mathrm{low}}}\sum_{t=0}^{D-1}(1-\alpha\mu)^{2t} \\
&= (1-\alpha\mu)^{2D}\mathbb{E}\|Y_k^0 - Y^\star(X_k)\|_{\mathrm{F}}^2 + \frac{\alpha^2\sigma^2[1-(1-\alpha\mu)^{2D}]}{B_{\mathrm{low}}[1-(1-\alpha\mu)^2]}. \qquad (21)
\end{aligned}$$

If $\alpha\mu = 1$, then

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|Y_k^D - Y^\star(X_k)\|_{\mathrm{F}}^2 \le \frac{\alpha^2\sigma^2}{B_{\mathrm{low}}} = \frac{\alpha\sigma^2}{\mu B_{\mathrm{low}}} \le \frac{2\alpha\sigma^2}{\mu B_{\mathrm{low}}},$$

which proves both (20) and (22). Suppose $0 < \alpha\mu < 1$. For $k \ge 1$, Young's inequality with weight $[1-(1-\alpha\mu)^D]/(1-\alpha\mu)^D$, warm starting, and (14) give

$$\begin{aligned}
&(1-\alpha\mu)^{2D}\mathbb{E}\|Y_k^0 - Y^\star(X_k)\|_{\mathrm{F}}^2 \\
&\le (1-\alpha\mu)^D\mathbb{E}\|Y_{k-1}^D - Y^\star(X_{k-1})\|_{\mathrm{F}}^2 + \frac{(1-\alpha\mu)^{2D}}{1-(1-\alpha\mu)^D}\mathbb{E}\|Y^\star(X_{k-1}) - Y^\star(X_k)\|_{\mathrm{F}}^2 \\
&\le (1-\alpha\mu)^D\mathbb{E}\|Y_{k-1}^D - Y^\star(X_{k-1})\|_{\mathrm{F}}^2 + \frac{(1-\alpha\mu)^{2D}L^2}{[1-(1-\alpha\mu)^D]\mu^2}\mathbb{E}\|X_k - X_{k-1}\|_{\mathrm{F}}^2.
\end{aligned}$$

Summing (21), including $Y_0^0 = Y_0$, gives

$$\begin{aligned}
&\sum_{k=0}^{T-1}\mathbb{E}\|Y_k^D - Y^\star(X_k)\|_{\mathrm{F}}^2 \\
&\le (1-\alpha\mu)^{2D}\|Y_0 - Y^\star(X_0)\|_{\mathrm{F}}^2 + (1-\alpha\mu)^D\sum_{k=0}^{T-2}\mathbb{E}\|Y_k^D - Y^\star(X_k)\|_{\mathrm{F}}^2 \\
&\quad + \frac{(1-\alpha\mu)^{2D}L^2}{[1-(1-\alpha\mu)^D]\mu^2}\sum_{k=0}^{T-2}\mathbb{E}\|X_{k+1} - X_k\|_{\mathrm{F}}^2 + \frac{T\alpha^2\sigma^2[1-(1-\alpha\mu)^{2D}]}{B_{\mathrm{low}}[1-(1-\alpha\mu)^2]}.
\end{aligned}$$

Since the omitted error at $T-1$ is nonnegative, rearrangement yields

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|Y_k^D - Y^\star(X_k)\|_{\mathrm{F}}^2 \le \frac{(1-\alpha\mu)^{2D}\|Y_0 - Y^\star(X_0)\|_{\mathrm{F}}^2}{T[1-(1-\alpha\mu)^D]} + \frac{\alpha^2\sigma^2[1+(1-\alpha\mu)^D]}{B_{\mathrm{low}}[1-(1-\alpha\mu)^2]} + \frac{(1-\alpha\mu)^{2D}L^2}{[1-(1-\alpha\mu)^D]^2\mu^2 T}\sum_{k=0}^{T-2}\mathbb{E}\|X_{k+1}-X_k\|_{\mathrm{F}}^2.$$

Finally,

$$1-(1-\alpha\mu)^D \ge \alpha\mu > 0,$$
$$\frac{(1-\alpha\mu)^{2D}}{1-(1-\alpha\mu)^D} \le \frac{1}{\alpha\mu},$$
$$\frac{(1-\alpha\mu)^{2D}L^2}{[1-(1-\alpha\mu)^D]^2\mu^2} \le \frac{L^2}{\alpha^2\mu^4},$$
$$\frac{\alpha^2\sigma^2[1+(1-\alpha\mu)^D]}{B_{\mathrm{low}}[1-(1-\alpha\mu)^2]} = \frac{\alpha\sigma^2[1+(1-\alpha\mu)^D]}{\mu B_{\mathrm{low}}(2-\alpha\mu)} \le \frac{2\alpha\sigma^2}{\mu B_{\mathrm{low}}}.$$

These bounds first give the movement-dependent estimate

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|Y_k^D - Y^\star(X_k)\|_{\mathrm{F}}^2 \le \frac{\|Y_0 - Y^\star(X_0)\|_{\mathrm{F}}^2}{\alpha\mu T} + \frac{2\alpha\sigma^2}{\mu B_{\mathrm{low}}} + \frac{L^2}{\alpha^2\mu^4 T}\sum_{k=0}^{T-2}\mathbb{E}\|X_{k+1}-X_k\|_{\mathrm{F}}^2. \quad (22)$$

For exact polar, $\|O_k\|_{\mathrm{F}}^2 = \mathrm{rank}(M_k)$; for finite NS, $\|O_k\|_{\mathrm{F}}^2 \le \mathrm{rank}(M_k)$. Hence $\|X_{k+1}-X_k\|_{\mathrm{F}}^2 = \beta^2\|O_k\|_{\mathrm{F}}^2 \le \beta^2\,\mathrm{rank}(M_k) \le \beta^2 r$. Substituting into (22) and using $(T-1)/T \le 1$ proves (20). We retain (22) for the joint tracking argument. Empty sums are zero. For finite T, bounded UL increments for either spectral update, (14), finite initialization, and the conditional noise bound imply finite second moments by the displayed recursion. □

Lemma 6 (Hypergradient-momentum tracking)**.** *In the expressions below, $\mathbb{E}\widehat{\nabla\Phi}(X_k)$ averages only the fresh hypergradient batches with X_k, Y_k^D fixed, as in Lemma 4; an outer expectation also averages the iterates. For $0<\eta<1/L$ and $0\le\theta<1$, $\|M_k\|_{\mathrm{F}} \le \left(C_f + \frac{LC_f}{\mu}\right)$, and*

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_{\mathrm{F}}^2 \le \frac{4\left(C_f+\frac{LC_f}{\mu}\right)^2}{T(1-\theta)} + \frac{6L_\Phi^2}{T(1-\theta)^2}\sum_{k=0}^{T-2}\mathbb{E}\|X_{k+1}-X_k\|_{\mathrm{F}}^2 + \frac{6}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\mathbb{E}\widehat{\nabla\Phi}(X_k) - \nabla\Phi(X_k)\|_{\mathrm{F}}^2 + (1-\theta)\left(C_f+\frac{LC_f}{\mu}\right)^2. \quad (23)$$

Proof. The sample derivative bounds and (18) imply

$$\|\widehat{\nabla\Phi}(X_k)\|_{\mathrm{F}} \le \|\nabla_X F(\mathcal{D}_{\mathrm{F}})\|_{\mathrm{F}} + \|\nabla_{XY}^2 G(\mathcal{D}_G)\|_{\mathrm{op}}\|v_Q\|_{\mathrm{F}} \le C_f + \frac{LC_f}{\mu} \quad \text{almost surely.} \quad (24)$$

By $M_{-1}=0$, (24), and induction, $\|M_k\|_{\mathrm{F}} \le \theta\|M_{k-1}\|_{\mathrm{F}} + (1-\theta)\|\widehat{\nabla\Phi}(X_k)\|_{\mathrm{F}} \le \left(C_f+\frac{LC_f}{\mu}\right)$. For $k\ge 1$, subtracting the current hypergradient from the momentum recursion gives

$$M_k - \nabla\Phi(X_k) = \theta[M_{k-1} - \nabla\Phi(X_{k-1})] + \theta[\nabla\Phi(X_{k-1}) - \nabla\Phi(X_k)] + (1-\theta)[\mathbb{E}\widehat{\nabla\Phi}(X_k) - \nabla\Phi(X_k)] + (1-\theta)[\widehat{\nabla\Phi}(X_k) - \mathbb{E}\widehat{\nabla\Phi}(X_k)].$$

Condition on all iterates and samples generated before the fresh UL batches at iteration k. The first three terms are then fixed. By fresh sampling, the conditional mean of the estimator is $\mathbb{E}\widehat{\nabla\Phi}(X_k)$, so its centered noise has zero cross term with their sum. Taking total expectations gives

$$\begin{aligned}\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_{\mathrm{F}}^2 = \mathbb{E}\Big\|&\theta[M_{k-1} - \nabla\Phi(X_{k-1})] + \theta[\nabla\Phi(X_{k-1}) - \nabla\Phi(X_k)] \\ &+ (1-\theta)[\mathbb{E}\widehat{\nabla\Phi}(X_k) - \nabla\Phi(X_k)]\Big\|_{\mathrm{F}}^2 \\ &+ (1-\theta)^2\mathbb{E}\|\widehat{\nabla\Phi}(X_k) - \mathbb{E}\widehat{\nabla\Phi}(X_k)\|_{\mathrm{F}}^2.\end{aligned}$$

Young's inequality with weight $(1-\theta)/2$, followed by $\|U+V\|_{\mathrm{F}}^2 \le 2\|U\|_{\mathrm{F}}^2 + 2\|V\|_{\mathrm{F}}^2$, gives

$$\begin{aligned}&\Big\|\theta[M_{k-1} - \nabla\Phi(X_{k-1})] + \theta[\nabla\Phi(X_{k-1}) - \nabla\Phi(X_k)] \\ &\qquad + (1-\theta)[\mathbb{E}\widehat{\nabla\Phi}(X_k) - \nabla\Phi(X_k)]\Big\|_{\mathrm{F}}^2 \\ &\le \left(1 + \frac{1-\theta}{2}\right)\theta^2\|M_{k-1} - \nabla\Phi(X_{k-1})\|_{\mathrm{F}}^2 \\ &\quad + 2\left(1 + \frac{2}{1-\theta}\right)\theta^2\|\nabla\Phi(X_{k-1}) - \nabla\Phi(X_k)\|_{\mathrm{F}}^2 \\ &\quad + 2\left(1 + \frac{2}{1-\theta}\right)(1-\theta)^2\|\mathbb{E}\widehat{\nabla\Phi}(X_k) - \nabla\Phi(X_k)\|_{\mathrm{F}}^2.\end{aligned}$$

For $0 \le \theta < 1$,

$$\begin{aligned}&\left(1 + \frac{1-\theta}{2}\right)\theta^2 \le \theta, \\ &2\left(1 + \frac{2}{1-\theta}\right)\theta^2 \le 2 + \frac{4}{1-\theta} \le \frac{6}{1-\theta}, \\ &2\left(1 + \frac{2}{1-\theta}\right)(1-\theta)^2 = 2(1-\theta)^2 + 4(1-\theta) \le 6(1-\theta).\end{aligned}$$

For the next display, condition on X_k, Y_k^D and take all expectations only over the fresh hypergradient batches. The variance identity and (24) give

$$\mathbb{E}[\|\widehat{\nabla\Phi}(X_k) - \mathbb{E}\widehat{\nabla\Phi}(X_k)\|_{\mathrm{F}}^2] = \mathbb{E}[\|\widehat{\nabla\Phi}(X_k)\|_{\mathrm{F}}^2] - \|\mathbb{E}\widehat{\nabla\Phi}(X_k)\|_{\mathrm{F}}^2 \le \left(C_f + \frac{LC_f}{\mu}\right)^2. \tag{25}$$

Using (13) and taking total expectations in (25),

$$\begin{aligned}&\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_{\mathrm{F}}^2 \le \theta\mathbb{E}\|M_{k-1} - \nabla\Phi(X_{k-1})\|_{\mathrm{F}}^2 + \frac{6L_\Phi^2}{1-\theta}\mathbb{E}\|X_k - X_{k-1}\|_{\mathrm{F}}^2 \\ &+ 6(1-\theta)\mathbb{E}\|\mathbb{E}\widehat{\nabla\Phi}(X_k) - \nabla\Phi(X_k)\|_{\mathrm{F}}^2 \\ &+ (1-\theta)^2\left(C_f + \frac{LC_f}{\mu}\right)^2, \\ &\mathbb{E}\|M_0 - \nabla\Phi(X_0)\|_{\mathrm{F}}^2 \le 4\left(C_f + \frac{LC_f}{\mu}\right)^2.\end{aligned}$$

Summing for $1 \le k < T$, adding the initial error, and extending nonnegative sums,

$$\begin{aligned}\sum_{k=0}^{T-1}\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_{\mathrm{F}}^2 &\le 4\left(C_f + \frac{LC_f}{\mu}\right)^2 + \theta\sum_{k=0}^{T-1}\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_{\mathrm{F}}^2 \\ &+ \frac{6L_\Phi^2}{1-\theta}\sum_{k=0}^{T-2}\mathbb{E}\|X_{k+1} - X_k\|_{\mathrm{F}}^2 \\ &+ 6(1-\theta)\sum_{k=0}^{T-1}\mathbb{E}\|\mathbb{E}\widehat{\nabla\Phi}(X_k) - \nabla\Phi(X_k)\|_{\mathrm{F}}^2\end{aligned}$$

$$+ T(1-\theta)^2 \left(C_f + \frac{LC_f}{\mu} \right)^2 .$$

Rearranging and dividing by $T(1-\theta) > 0$ gives (23). The argument also includes $\theta = 0$. □

Proof of Lemma 1.

For exact polar, the supported singular values of O_k are all one. For finite NS, they lie in $(0, 1]$, with no new singular directions, as shown in the proof of Lemma 9. Both maps send zero to zero. Thus $\|Y^\star(X_{k+1}) - Y^\star(X_k)\|_{\mathrm{F}}^2 \le \frac{L^2\beta^2}{\mu^2}\|O_k\|_{\mathrm{F}}^2 \le \frac{L^2\beta^2}{\mu^2}\operatorname{rank}(M_k)$, where the first inequality uses the LL-solution Lipschitz bound (14). Applying the warm-started LL tracking bound (22) with $X_{k+1} - X_k = -\beta O_k$ gives (6); retaining $\|O_k\|_{\mathrm{F}}^2$ gives the corresponding movement bound.

Substitute the conditional hypergradient bias bound (19) into the momentum bound (23). The LL-error contribution is $12L_\Phi^2 T^{-1}\sum_{k=0}^{T-1}\mathbb{E}\|Y_k^D - Y^\star(X_k)\|_{\mathrm{F}}^2$. Applying the movement-dependent bound in (22) to this term gives the bound below. The first movement coefficient comes from changes in the true hypergradient; the second comes from LL-solution tracking. The rank identities above give the remaining claims.

The same argument also gives the following momentum bound used in the convergence proof:

$$\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_{\mathrm{F}}^2 \\
&\le \frac{4(C_f + LC_f/\mu)^2}{T(1-\theta)} + (1-\theta)(C_f + LC_f/\mu)^2 + \frac{12L_\Phi^2\|Y_0 - Y^\star(X_0)\|_{\mathrm{F}}^2}{\alpha\mu T} \\
&\quad + \frac{24L_\Phi^2\alpha\sigma^2}{\mu B_{\mathrm{low}}} + \frac{12L^2{C_f}^2}{\mu^2}(1-\eta\mu)^{2Q+2} \\
&\quad + \frac{\beta^2}{T}\left[\frac{6L_\Phi^2}{(1-\theta)^2} + \frac{12L_\Phi^2L^2}{\alpha^2\mu^4}\right]\sum_{k=0}^{T-2}\mathbb{E}\|O_k\|_{\mathrm{F}}^2. \qquad (26)
\end{aligned}$$

For exact polar, $\|O_k\|_{\mathrm{F}}^2 = \operatorname{rank}(M_k)$, so the last sum is exactly $\sum_{k=0}^{T-2}\mathbb{E}\operatorname{rank}(M_k)$. For finite NS, $\|O_k\|_{\mathrm{F}}^2 \le \operatorname{rank}(M_k)$, and replacing the last sum by the same rank sum remains valid. Empty sums are zero.

□

Lemma 7 (Tracking under the Theorem 1 stepsize schedule)**.** *Suppose Assumptions 1–3 hold. Choose the parameters of Theorem 1 and use either UL update. Then*

$$\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_{\mathrm{F}}^2 \\
&\le T^{-1/2}\Bigg[5\left(C_f + \frac{LC_f}{\mu}\right)^2 + \frac{12L_\Phi^2 L}{\mu}\|Y_0 - Y^\star(X_0)\|_{\mathrm{F}}^2 + \frac{24L_\Phi^2\sigma^2}{L\mu B_{\mathrm{low}}} \\
&\qquad + \frac{12L^2{C_f}^2}{\mu^2} + r\beta_0^2 L_\Phi^2\left(6 + \frac{12L^4}{\mu^4}\right)\Bigg]. \qquad (27)
\end{aligned}$$

Proof. By orthonormality of the support singular vectors, $\|\operatorname{polar}(M_k)\|_{\mathrm{op}} \le 1$, and $\|\operatorname{polar}(M_k)\|_{\mathrm{F}}^2 = \operatorname{rank}(M_k) \le r$. If $M_k = 0$, both norms are zero. Lemma 9 gives $\|\operatorname{NS}(M_k, q)\|_{\mathrm{op}} \le 1$, hence $\|\operatorname{NS}(M_k, q)\|_{\mathrm{F}}^2 \le r$. Thus, for either update, $\|X_{k+1} - X_k\|_{\mathrm{F}}^2 = \beta^2\|O_k\|_{\mathrm{F}}^2 \le r\beta^2$, $\frac{1}{T}\sum_{k=0}^{T-2}\mathbb{E}\|X_{k+1} - X_k\|_{\mathrm{F}}^2 \le \frac{T-1}{T}r\beta^2 \le r\beta^2$. The schedule satisfies $0 < \alpha \le 1/L$, $0 < \eta < 1/L$, $0 \le \theta < 1$, and $D \ge 1$. Thus (26) applies to either update. Bound its final sum by $(T-1)r \le Tr$ and substitute the schedule below. All these expectations are finite by the LL recursion, bounded derivative estimates, and the bounded increments above. Direct

substitution yields

$$\frac{1}{T(1-\theta)} = 1-\theta = T^{-1/2}, \qquad \frac{1}{\alpha\mu T} = \frac{L}{\mu}T^{-1/2},$$
$$\alpha = L^{-1}T^{-1/2}, \qquad (1-\eta\mu)^{2Q+2} \le (1-\eta\mu)^{2Q} \le T^{-1/2},$$
$$\frac{r\beta^2}{(1-\theta)^2} = r\beta_0^2 T^{-1/2}, \qquad \frac{r\beta^2}{\alpha^2} = r\beta_0^2 L^2 T^{-1/2}.$$

Substituting these bounds gives (27). □

Lemma 8 (One-step progress of exact-polar BiMuon)**.** *For the exact-polar update* $X_{k+1} = X_k - \beta\,\mathrm{polar}(M_k)$ *with* $\beta > 0$*,*

$$\mathbb{E}\|\nabla\Phi(X_k)\|_* \le \frac{\mathbb{E}\Phi(X_k) - \mathbb{E}\Phi(X_{k+1})}{\beta} + 2\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_* + \frac{L_\Phi r\beta}{2}. \tag{28}$$

Proof. By the smoothness of Φ and $\|\mathrm{polar}(M_k)\|_{\mathrm{F}}^2 \le r$,

$$\begin{aligned}\Phi(X_{k+1}) &\le \Phi(X_k) - \beta\langle\nabla\Phi(X_k), \mathrm{polar}(M_k)\rangle + \frac{L_\Phi r\beta^2}{2}\\ &= \Phi(X_k) - \beta\|M_k\|_* + \beta\langle M_k - \nabla\Phi(X_k), \mathrm{polar}(M_k)\rangle + \frac{L_\Phi r\beta^2}{2}\\ &\le \Phi(X_k) - \beta\|\nabla\Phi(X_k)\|_* + 2\beta\|M_k - \nabla\Phi(X_k)\|_* + \frac{L_\Phi r\beta^2}{2}.\end{aligned}$$

The equality uses $\langle M_k, \mathrm{polar}(M_k)\rangle = \|M_k\|_*$. The last inequality follows from spectral–nuclear duality, $\|\mathrm{polar}(M_k)\|_{\mathrm{op}} \le 1$, and the triangle inequality. Rearranging and taking expectations proves (28). □

Proof of Theorem 1: exact-polar case.

Averaging (28) over $k = 0, \dots, T-1$ and using $\mathbb{E}\Phi(X_T) \ge \Phi_{\mathrm{inf}}$ gives

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\nabla\Phi(X_k)\|_* \le \frac{\Phi(X_0) - \Phi_{\mathrm{inf}}}{\beta T} + \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_* + \frac{L_\Phi r\beta}{2}.$$

For every matrix $A \in \mathbb{R}^{m\times n}$, Cauchy–Schwarz on its singular values gives $\|A\|_* \le \sqrt{r}\|A\|_{\mathrm{F}}$. Applying Cauchy–Schwarz first in probability and then to the finite sum,

$$\begin{aligned}\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_* &\le \frac{\sqrt{r}}{T}\sum_{k=0}^{T-1}\left(\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_{\mathrm{F}}^2\right)^{1/2}\\ &\le \left[\frac{r}{T}\sum_{k=0}^{T-1}\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_{\mathrm{F}}^2\right]^{1/2}.\end{aligned}$$

Using Lemma 7, $1/(\beta T) = \beta_0^{-1}T^{-1/4}$, and $T^{-3/4} \le T^{-1/4}$ proves the bound with

$$\begin{aligned}C_{\mathrm{pol}} = \frac{\Phi(X_0) - \Phi_{\mathrm{inf}}}{\beta_0} + \frac{L_\Phi r\beta_0}{2}\\ + 2\Bigg\{r\Bigg[5\left(C_f + \frac{LC_f}{\mu}\right)^2 + \frac{12L_\Phi^2 L}{\mu}\|Y_0 - Y^\star(X_0)\|_{\mathrm{F}}^2 + \frac{24L_\Phi^2\sigma^2}{L\mu B_{\mathrm{low}}}\\ + \frac{12L^2{C_f}^2}{\mu^2} + r\beta_0^2 L_\Phi^2\left(6 + \frac{12L^4}{\mu^4}\right)\Bigg]\Bigg\}^{1/2}.\end{aligned} \tag{29}$$

The constant is independent of T and is valid for every fixed $\beta_0 > 0$. Its three additive terms account for the initial UL objective gap, the smoothness cost of the UL updates, and the momentum tracking error bounded in Lemma 7, respectively. The expectations of objective values exist, since

$\|X_k - X_0\|_{\mathrm{F}} \le k\beta\sqrt{r}$, and $|\Phi(X_k) - \Phi(X_0)| \le (C_f + LC_f/\mu)k\beta\sqrt{r}$. Independence and uniformity of R give

$$\mathbb{E}\|\nabla\Phi(X_R)\|_* = \sum_{k=0}^{T-1} \mathbb{P}(R=k)\mathbb{E}\|\nabla\Phi(X_k)\|_* = \frac{1}{T}\sum_{k=0}^{T-1} \mathbb{E}\|\nabla\Phi(X_k)\|_*.$$

This proves Theorem 1(i). Choosing $T \ge \max\{2, \lceil (C_{\mathrm{pol}}/\epsilon)^4 \rceil\}$ then ensures $\mathbb{E}\|\nabla\Phi(X_R)\|_* \le \epsilon$. □

D Convergence proof for finite-NS BiMuon

We follow the polar-error descent argument in Appendix B of Kim & Oh (2026), using the bilevel tracking bound established under the schedule of Theorem 1. The Frobenius smoothness used here contributes the factor r in the quadratic remainder.

The finite-NS map and its implementation are given in Appendix B.

Lemma 9 (One-step progress of finite-NS BiMuon)**.** *Suppose Assumptions 1–3 hold. For Algorithm 1 with $0 < \eta < 1/L$, $0 \le \theta < 1$, $\beta > 0$, an integer $q \ge 0$, and $O_k = \mathrm{NS}(M_k, q)$,*

$$\|O_k\|_{\mathrm{op}} \le 1, \qquad \|O_k\|_{\mathrm{F}}^2 \le \mathrm{rank}(M_k) \le r. \tag{30}$$

If (7) *holds, then*

$$(1-\varepsilon_q)\mathbb{E}\|\nabla\Phi(X_k)\|_* \le \frac{\mathbb{E}\Phi(X_k) - \mathbb{E}\Phi(X_{k+1})}{\beta} + 2\mathbb{E}\|M_k - \nabla\Phi(X_k)\|_* + \frac{L_\Phi r\beta}{2}. \tag{31}$$

Proof. For the Taylor polynomial defined in Appendix B, Proposition 1 of Kim & Oh (2026) establishes preservation of the unit spectral ball. In scalar form, the properties needed here are $1 \le p(x) \le p(0) \quad (0 \le x \le 1)$, and $p(x) \le x^{-1/2} \quad (0 < x \le 1)$. The latter also follows directly by bounding the finite Taylor sum by the full series for $x^{-1/2}$. Each positive initial singular value of $O^{[0]}$ is at most one; the NS recurrence sends s to $sp(s^2) \in [s, 1]$, while zero stays zero. This proves (30), including zero and rank-deficient momentum. The NS map is continuous and hence measurable.

Insert the polar factor as in the single-level proof:

$$\begin{aligned}
\langle M_k, O_k\rangle &= \|M_k\|_* + \langle M_k, O_k - \mathrm{polar}(M_k)\rangle \ge (1-\varepsilon_q)\|M_k\|_*, \\
\langle \nabla\Phi(X_k), O_k\rangle &\ge (1-\varepsilon_q)\|M_k\|_* - \|M_k - \nabla\Phi(X_k)\|_* \\
&\ge (1-\varepsilon_q)\|\nabla\Phi(X_k)\|_* - (2-\varepsilon_q)\|M_k - \nabla\Phi(X_k)\|_* \\
&\ge (1-\varepsilon_q)\|\nabla\Phi(X_k)\|_* - 2\|M_k - \nabla\Phi(X_k)\|_*.
\end{aligned}$$

The first two inequalities use spectral–nuclear duality and the last two use the triangle inequality and $0 \le \varepsilon_q < 1$. Applying (16) gives $\Phi(X_{k+1}) \le \Phi(X_k) - \beta(1-\varepsilon_q)\|\nabla\Phi(X_k)\|_* + 2\beta\|M_k - \nabla\Phi(X_k)\|_* + \frac{L_\Phi r\beta^2}{2}$. Rearranging and taking expectations yields (31). These expectations are finite: the updates have norm at most $\beta\sqrt{r}$, and both the true hypergradient and the momentum are bounded by $C_f + LC_f/\mu$. □

Polar approximation. Lemma 2 restates the Taylor NS residual and polar-error bounds of Kim & Oh (2026, Theorem 2, Lemmas 1–3, and Corollary 5).

Let $O_k^{[j]}$ denote the jth inner iterate of Algorithm 2 at outer iteration k, and let Π_k be the orthogonal projector onto the column space of M_k. Define $\delta_{k,j} := \|\Pi_k - O_k^{[j]}(O_k^{[j]})^\top\|_{\mathrm{op}}$; when $M_k = 0$, take $\Pi_k = 0$, so this residual is zero. For the degree-d Taylor iteration, the residual satisfies $\delta_{k,q} \le \delta_{k,0}^{(d+1)^q}$. Under the uniform initial bound in Lemma 2, the resulting approximation factor satisfies $\frac{1}{1-\varepsilon_q} \le \frac{1}{\sqrt{1-\delta_0^{(d+1)^q}}}$ and approaches one doubly exponentially with q for fixed $\delta_0 < 1$. We apply these bounds on the positive singular support; Algorithm 2 places its initial singular values in $(0, 1]$ and preserves that support. Zero input is mapped to zero. The cited result gives $\mathrm{NS}(M, q) \to \mathrm{polar}(M)$ for each fixed input M, but this pointwise conclusion does not imply the deterministic run-wise error condition (7) used in Theorem 1.

Proof of Theorem 1: finite-NS case.

The schedule in Theorem 1 satisfies the hypotheses of Lemma 7. Averaging (31) and using $\mathbb{E}\Phi(X_T) \geq \Phi_{\inf}$ gives

$$\begin{aligned}
&(1-\varepsilon_q)\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\nabla\Phi(X_k)\|_* \\
&\leq \frac{\Phi(X_0)-\Phi_{\inf}}{\beta T} + \frac{2}{T}\sum_{k=0}^{T-1}\mathbb{E}\|M_k-\nabla\Phi(X_k)\|_* + \frac{L_\Phi r\beta}{2} \\
&\leq \frac{\Phi(X_0)-\Phi_{\inf}}{\beta_0}T^{-1/4} + 2\left[\frac{r}{T}\sum_{k=0}^{T-1}\mathbb{E}\|M_k-\nabla\Phi(X_k)\|_{\mathrm{F}}^2\right]^{1/2} + \frac{L_\Phi r\beta_0}{2}T^{-3/4} \\
&\leq C_{\mathrm{pol}}T^{-1/4}.
\end{aligned}$$

The second inequality uses $\|A\|_* \leq \sqrt{r}\|A\|_{\mathrm{F}}$ and Cauchy–Schwarz in probability and over the finite sum. The last inequality follows from (27) and (29). Division by the positive deterministic $1-\varepsilon_q$ and independent uniform selection of R prove Theorem 1(ii). No unbiased-hypergradient assumption is introduced: LL-solution and Neumann bias are already included in Lemma 7. □

Uniform approximation condition. Theorem 1(ii) assumes a deterministic bound $\varepsilon_q < 1$ for the specified run. A fixed NS depth alone does not guarantee that this bound remains uniform as T varies under the capped normalization in Algorithm 2. Thus, the horizon-uniform rate requires a common $\varepsilon_q < 1$ independent of T.

Computational complexity. A LL mini-batch mean uses B_{low} individual sample gradients per step. Each outer iteration performs D such steps. Algorithm 3 uses exactly Q Hessian batches of size B_H, each applied to one recursively computed matrix; it includes $Q+1$ terms but does not require $Q+1$ Hessian batches. The UL loss supplies two partial derivatives from a batch of size B_f. The mixed action uses B_{xy} samples. For exact polar, or NS with a common polar-error bound below one, the derivative counts are as follows. The orders in the final column hold with that error bound and the problem parameters fixed:

Operation	Count through T iterations	Order at nuclear accuracy ϵ
Lower $\nabla_Y G$	TDB_{low}	$O(\epsilon^{-4})$
Upper partial $\nabla_X F$	TB_f	$O(\epsilon^{-4})$
Upper partial $\nabla_Y F$	TB_f	$O(\epsilon^{-4})$
Hessian-vector actions	TQB_H	$O(\epsilon^{-4}\log(1/\epsilon))$
Mixed derivative actions	TB_{xy}	$O(\epsilon^{-4})$

The table counts derivative evaluations separately. Finite NS additionally requires q matrix-multiplication iterations per UL update, while exact polar requires a matrix decomposition. These matrix-arithmetic costs are separate from the stochastic derivative complexity reported above.

E Proofs and Details for Blockwise Matrix BiMuon

This appendix proves the blockwise LL sensitivity bound and the relative stepsize allocation rule in Section 6, and provides additional details on the joint sensitivity condition and its finite-D interpretation.

Proposition 2 (Blockwise sensitivity of the LL solution)**.** *Suppose that $g(X,\cdot)$ is differentiable and μ-strongly convex for every X, and that, for constants $L_{XY,b} > 0$, the joint blockwise Lipschitz condition*

$$\|\nabla_Y g(X,Y) - \nabla_Y g(X',Y)\|_{\mathrm{F}}^2 \leq \sum_{b=1}^{J} L_{XY,b}^2 \|X^{(b)} - X'^{(b)}\|_{\mathrm{F}}^2 \tag{32}$$

holds for all X, X', Y. Then

$$\|Y^\star(X) - Y^\star(X')\|_{\mathrm{F}}^2 \le \frac{1}{\mu^2} \sum_{b=1}^{J} L_{XY,b}^2 \left\| X^{(b)} - X'^{(b)} \right\|_{\mathrm{F}}^2 . \tag{33}$$

Proof of Proposition 2.

Differentiability and strong convexity on the full LL Euclidean space give a unique minimizer: the strong-convexity quadratic lower bound is coercive, so continuity gives existence, and strong convexity gives uniqueness. The two LL gradients vanish at their respective minimizers. Strong monotonicity at fixed X and Frobenius Cauchy–Schwarz give

$$\begin{aligned}\mu\|Y^\star(X) - Y^\star(X')\|_{\mathrm{F}}^2 &\le \langle \nabla_Y g(X, Y^\star(X')) - \nabla_Y g(X, Y^\star(X)), Y^\star(X') - Y^\star(X)\rangle \\ &\le \|\nabla_Y g(X, Y^\star(X')) - \nabla_Y g(X', Y^\star(X'))\|_{\mathrm{F}} \|Y^\star(X) - Y^\star(X')\|_{\mathrm{F}}.\end{aligned}$$

If the minimizers coincide, (33) is immediate. Otherwise divide by their positive distance, square, and apply (32). Substitution of the update in (8) gives the intermediate bound $\|Y^\star(X_{k+1}) - Y^\star(X_k)\|_{\mathrm{F}}^2 \le \frac{\beta^2}{\mu^2} \sum_{b=1}^{J} L_{XY,b}^2 \omega_{b,k}^2 \|O_k^{(b)}\|_{\mathrm{F}}^2$. For exact polar, the squared norm equals the supported rank. For the Taylor NS map, the singular values start in $[0, 1]$, remain there under the specified polynomial, and no new supported direction appears; hence the squared norm is at most that rank, including zero momentum. This is the same bound used in Lemma 9. Bounding each rank by r_b and using $w_b = L_{XY,b}^2 r_b/\mu^2$ yields the LL target-movement bound

$$\|Y^\star(X_{k+1}) - Y^\star(X_k)\|_{\mathrm{F}}^2 \le \beta^2 \sum_{b=1}^{J} w_b \omega_{b,k}^2. \tag{34}$$

□

At an outer transition, warm starting gives $Y_{k+1}^0 = Y_k^D$. For any $t > 0$, Young's inequality gives

$$\|Y_k^D - Y^\star(X_{k+1})\|_{\mathrm{F}}^2 \le (1+t)\|Y_k^D - Y^\star(X_k)\|_{\mathrm{F}}^2 + (1+t^{-1})\|Y^\star(X_{k+1}) - Y^\star(X_k)\|_{\mathrm{F}}^2.$$

Under the same LL contraction/noise conditions used in Lemma 5, the finite-D recursion contracts this new initial error in conditional expectation and adds its sampling term. Equation (34) controls the target-movement contribution; it does not imply monotonicity of the realized error because the two vector errors may reinforce or cancel.

Proof of the allocation rule in (10)**.** For $a_{b,k} \ge 0$ and fixed $w_b > 0$, weighted Cauchy–Schwarz yields $\sum_b a_{b,k}\omega_{b,k} \le \sqrt{\sum_b a_{b,k}^2/w_b}\sqrt{\sum_b w_b\omega_{b,k}^2} \le \sqrt{\sum_b a_{b,k}^2/w_b}$. When some score is positive, equality requires a saturated constraint and proportionality between $\sqrt{w_b}\omega_{b,k}$ and $a_{b,k}/\sqrt{w_b}$, giving (10) and uniqueness. All-zero scores make every feasible allocation optimal; zero is a convenient choice. For exact polar, $\langle M, \mathrm{polar}(M)\rangle = \|M\|_* \ge 0$. Taylor NS has the same supported singular vectors and nonnegative singular values, so its score is also nonnegative.

With exactly the weights $w_b = L_{XY,b}^2 r_b/\mu^2$, equation (34) becomes $\|Y^\star(X_{k+1}) - Y^\star(X_k)\|_{\mathrm{F}}^2 \le \beta^2 \sum_b w_b\omega_{b,k}^2 \le \beta^2$ under the constraint in (9). Since $\beta > 0$, taking square roots proves the stated drift bound. Positivity of $L_{XY,b}$, r_b and μ ensures $w_b > 0$.

F Experimental Details and Supplementary Results

This appendix provides the experimental details for the two real-data experiments in Section 7, followed by two supplementary controlled experiments that further examine the corresponding algorithmic designs.

F.1 CIFAR-10 Representation Learning: Experimental Details

Setup. The experiment in Section 7.1 uses a fixed class-balanced pool of 1,500 CIFAR-10 (Krizhevsky, 2009) training images, sampled with pool seed 20270921. A frozen ResNet-18 (He et al., 2016) pretrained on ImageNet (Deng et al., 2009), with its final classifier removed, provides 512-dimensional features using the preprocessing associated with torchvision's `IMAGENET1K_V1` weights. For each problem seed s, a stratified split assigns 100 images per class to LL training and 50 images per class to UL validation, giving 1,000 LL-training and 500 UL-validation samples. The split generator uses seed $20000 + s$.

Features are divided by a single RMS scalar computed from the LL training features, without whitening, centering, or coordinatewise rescaling. The learned representation is $\tanh(zX^\top)$, $X \in \mathbb{R}^{256\times 512}$, followed by a linear classifier $Y \in \mathbb{R}^{10\times 256}$. The LL objective is the mean training cross-entropy plus $0.01\|Y\|_{\mathrm{F}}^2/2$, and the UL objective is the mean validation cross-entropy. We initialize $X_0 = \frac{0.1}{\sqrt{512}}G$, $Y_0 = M_{-1} = 0$, where G is a standard Gaussian matrix generated using seed $30000 + s$. All methods use the same split, initialization, and stochastic training stream for each problem seed.

Methods and hyperparameters. All methods use $D = 5$ warm-started LL SGD steps with $\alpha = 0.1$, $Q = 4$ stochastic Hessian–vector products (five Neumann terms) with $\eta = 0.2$, fresh mini-batches of size 64, and EMA coefficient $\theta = 0.9$. Each run performs 60 UL iterations in float64.

The scalar-normalized baselines in Section 7.1 floor their normalization denominators at 10^{-30}. Exact-polar BiMuon applies the supported polar factor to the stochastic hypergradient momentum, retaining singular values larger than $10^{-14}\sigma_{\max}$. NS-BiMuon uses the practical Frobenius-normalized cubic iteration $Z_0 = \frac{M_k}{\|M_k\|_{\mathrm{F}}}$, $Z_{j+1} = \frac{3}{2}Z_j - \frac{1}{2}Z_jZ_j^\top Z_j$, $j = 0, \ldots, 11$, with zero output for zero momentum and the smaller Gram matrix used in the implementation. Thus, the real-data experiment uses a practical Frobenius-normalized NS implementation rather than the capped normalization used in the theoretical analysis.

Every method uses the same stochastic derivative budget: 300 LL-gradient mini-batch evaluations, 60 UL partial-gradient pairs, 240 Hessian–vector products, and 60 mixed-derivative products. Evaluation refits and spectral diagnostics are performed separately and do not affect the training state.

Tuning and evaluation. Stepsizes are selected using discovery seeds 0 and 1. For each method, the initial candidates are calibrated to first-update Frobenius norms $\{0.025,\ 0.1,\ 0.4\}$, pooled over the two discovery seeds. The candidate range is extended by factors of four when the selected value lies on an outer boundary. The common candidate set also includes $\beta = 0.14792005232672834$. The selected stepsize minimizes the mean final optimized UL validation loss over the two discovery seeds, with ties resolved in favor of the smaller stepsize. All selected stepsizes are frozen before evaluation on seeds 20–24, with 60 UL iterations per run.

Optimized UL objective. At every logged X_k, including initialization and all 60 UL updates, the LL classifier is independently refit by L-BFGS (Liu & Nocedal, 1989). The refit uses at most 300 iterations, gradient tolerance 10^{-10}, change tolerance 10^{-14}, history size 50, and a strong-Wolfe line search. The optimized UL validation loss reported in Section 7.1 is the validation cross-entropy evaluated after this independent LL refit.

Figure 2(a) reports means over evaluation seeds 20–24. The inset compares exact-polar and NS-BiMuon over the final five iterations, with standard-error bands. Table 1 reports the final means and sample standard deviations.

Momentum-spectrum diagnostic. Figure 2(b) uses the raw EMA momentum M_k before the UL transformation. Let $\{\sigma_i\}$ denote its singular values and define $\tau = \max\{10^{-12}, 10^{-6}\sigma_{\max}\}$. The spectrum-imbalance statistic is $\frac{Q_{0.9}(\{\sigma_i\})}{\max\{Q_{0.1}(\{\sigma_i\}),\tau\}}$, where $Q_{0.9}$ and $Q_{0.1}$ denote the 90th and 10th percentiles, respectively. A larger value indicates a more unequal singular spectrum. For each method and iteration, Figure 2(b) reports the median across the five evaluation seeds, with first- and third-quartile bands. The diagnostic is computed on the input momentum M_k, rather than on the transformed UL direction.

Table 1: Frozen stepsizes and final optimized UL validation loss after 60 updates on CIFAR-10. Uncertainties are sample standard deviations over seeds 20–24.

UL update	β	Optimized UL validation loss
stocBiO+EMA	1.0101525	$.182175 \pm .019159$
OpNorm	.14792005	$.199307 \pm .127671$
FrobNorm	.14792005	$.189266 \pm .021947$
Exact-polar BiMuon	.14792005	$.033333 \pm .005441$
NS-BiMuon	.14792005	$.037494 \pm .007741$

Runtime protocol. Figure 2(c) reports a separate benchmark on an NVIDIA A100-SXM4-40GB using PyTorch 1.12.1+cu116, CUDA 11.6, float64, with TF32 disabled. The normalized cubic NS depth is fixed to 12 and exact-polar and NS-BiMuon use the common UL stepsize 0.14792005232672834.

Single-transformation timing uses 12 momentum snapshots collected at iterations 1, 30, and 60 from seeds 20 and 21. Both transformations receive the same input matrices, with five warmup calls followed by seven timed rounds of ten calls per method in randomized method order.

The 60-update timing uses seeds 20–24 with three repetitions per seed and includes the LL solver, stochastic hypergradient estimator, EMA update, batching, and UL transformation. Feature extraction, LL evaluation refits, spectral diagnostics, and logging are excluded. Reported speedups are medians of paired exact-polar/NS timing ratios. The marginal median runtimes are 16.91 ms versus 1.19 ms per transformation and 1.385 s versus 0.411 s over 60 updates for exact polar and NS-12, respectively.

F.2 CIFAR-100 Blockwise Allocation: Experimental Details

Setup. The experiment in Section 7.2 uses all 50,000 CIFAR-100 (Krizhevsky, 2009) training images. A fixed class-stratified split assigns 400 images per class to LL training and 100 images per class to UL validation, giving 40,000/10,000 samples, with split seed 20270922.

A frozen ResNet-50 (He et al., 2016) pretrained on ImageNet (Deng et al., 2009) with torchvision `IMAGENET1K_V2` weights provides 2,048-dimensional features. Preprocessing uses bilinear resizing to 232 pixels, a 224-pixel center crop, and channel normalization with means $(0.485, 0.456, 0.406)$ and standard deviations $(0.229, 0.224, 0.225)$. All features are divided by the LL-training RMS scalar 0.3188708678543272, without whitening, centering, or coordinatewise rescaling. All methods use the same cached features and fixed data split.

Controlled coupling and exact block sensitivity. Write $X, Y \in \mathbb{R}^{100\times 2048}$ as four column blocks $X^{(b)}, Y^{(b)} \in \mathbb{R}^{100\times 512}$, $b = 1, \dots, 4$. Set $c_b = \rho^{(b-1)/3}$, $C = \mathrm{blkdiag}(c_1 I_{512}, \dots, c_4 I_{512})$. For frozen features h_i and labels t_i, define $\ell_{\mathcal{S}}(Y) = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \mathrm{CE}(Y h_i, t_i)$. The bilevel objectives are $g(X,Y) = \ell_{\mathrm{tr}}(Y) + \frac{\mu}{2}\|Y - XC\|_{\mathrm{F}}^2$, $f(X,Y) = \ell_{\mathrm{val}}(Y)$, $\mu = 0.01$. The shared softmax cross-entropy couples all classifier blocks. Moreover, $\nabla^2_{XY} g(X,Y)[V] = -\mu VC$, and $\|\nabla_Y g(X,Y) - \nabla_Y g(X',Y)\|_{\mathrm{F}}^2 = \mu^2 \sum_{b=1}^4 c_b^2 \|X^{(b)} - X'^{(b)}\|_{\mathrm{F}}^2$. Hence $L_{XY,b} = \mu c_b$, $r_b = 100$, $w_b = 100 c_b^2$. The sensitivity ratio is $\rho = \frac{\max_b c_b}{\min_b c_b}$, and the corresponding largest-to-smallest ratio among the w_b is ρ^2.

Methods and hyperparameters. All runs start from $X_0 = Y_0 = M_{-1} = 0$ and use a seed-dependent mini-batch stream initialized with seed $100000 + s$. We use $D = 5$ warm-started LL updates, $Q = 4$ Hessian–vector products (five Neumann terms), batch size 64, and EMA coefficient $\theta = 0.9$ without bias correction.

The LL feature bound $L = \frac{1}{2}\max_{i\in\mathrm{tr}} \|h_i\|_2^2 + \mu$ determines $\alpha = \min\{0.1, 1/L\}$, $\eta = \min\{0.2, 1/L\}$, which both equal 0.0004690390185025686 in this experiment. These parameters are shared across all allocation rules and all values of ρ.

Each block uses the exact supported polar direction $O_k^{(b)} = \mathrm{polar}(M_k^{(b)})$, with singular values below $10^{-14}\sigma_{\max}$ removed. Let $\widetilde{\omega}_{b,k}$ denote the raw allocation weight. Every rule is normalized by

$\omega_{b,k} = \frac{\widetilde{\omega}_{b,k}}{\sqrt{\sum_c w_c \widetilde{\omega}_{c,k}^2}}$, followed by $X_{k+1}^{(b)} = X_k^{(b)} - \beta\omega_{b,k} O_k^{(b)}$. Positive raw weights are first divided by their maximum for numerical stability, which does not change the normalized allocation. If all progress scores vanish, the update is set to zero; otherwise $\sum_b w_b \omega_{b,k}^2 = 1$.

Each 60-iteration run uses 300 LL-gradient mini-batch evaluations, 240 Hessian–vector products, 60 mixed-derivative products, and 60 UL partial-gradient pairs. Optimization uses float64 with TF32 disabled on an NVIDIA A100-SXM4-40GB with PyTorch 2.11.0+cu128, torchvision 0.26.0+cu128, and CUDA 12.8. Frozen-backbone feature extraction is performed separately in float32.

Tuning and evaluation. A single common UL stepsize is selected from $\beta \in \{0.001, 0.01, 0.1, 1\}$ using 25-iteration discovery runs on seeds 31001 and 31002, all four allocation rules, and $\rho \in \{1, 10\}$. The selected stepsize minimizes the mean final optimized UL validation loss over the discovery runs, with ties resolved in favor of the smaller value. This gives $\beta = 0.1$, which is frozen for all final evaluations. Evaluation uses seeds 41001–41005, $\rho \in \{1, 3, 10\}$, and 60 UL iterations.

Optimized UL objective. At initialization and every five UL iterations, a deterministic full-data L-BFGS solve approximates $Y^\star(X)$, warm-started from the previous evaluation reference. The solve uses at most 300 iterations, learning rate 1, history size 30, strong-Wolfe line search, gradient tolerance 10^{-10}, change tolerance 10^{-14}, and full-objective accumulation in chunks of 10,000 images. A reference solution is accepted when $\|\nabla_Y g(X, \widehat{Y})\|_{\mathrm{F}} \le 10^{-5}$. The reported optimized UL loss is $\ell_{\mathrm{val}}(\widehat{Y})$.

Figure 3(a) reports means and sample-standard-deviation bands over the five evaluation seeds at $\rho = 10$. Figure 3(b) reports the final mean optimized UL loss at $\rho \in \{1, 3, 10\}$ with sample-standard-deviation error bars.

Table 2: Final optimized UL validation loss for CIFAR-100 blockwise allocation with common $\beta = 0.1$. Entries are means and sample standard deviations over five evaluation seeds.

ρ	Uniform	Progress-only	Sensitivity-only	Proposed
1	$.822789 \pm .002729$	$.822794 \pm .002733$	$.822789 \pm .002729$	$.822794 \pm .002733$
3	$.829264 \pm .002747$	$.841830 \pm .002435$	$.830485 \pm .002349$	$.822794 \pm .002733$
10	$.843067 \pm .002478$	$.860136 \pm .001990$	$.844844 \pm .001927$	$.822794 \pm .002733$

For both $\rho = 3$ and $\rho = 10$, the proposed allocation attains a lower final optimized UL loss than the three alternative allocation rules on all five evaluation seeds.

Scale invariance of the proposed rule. The nearly identical proposed results across different values of ρ follow from the scale structure of this controlled construction. Writing $Z^{(b)} = c_b X^{(b)}$, $M_k^{(b)} = c_b N_k^{(b)}$, and using $w_b = 100 c_b^2$, the factor c_b cancels from the effective proposed update in $Z^{(b)}$. Consequently, under the shared zero initialization and random stream, the proposed trajectories in (Z, Y) are invariant to ρ.

F.3 Supplementary Synthetic Representation Learning

Setup. We additionally study a controlled four-class nonlinear representation-learning problem with $X \in \mathbb{R}^{32\times 16}$, $Y \in \mathbb{R}^{4\times 32}$. The representation is $\tanh(zX^\top)$. The LL objective is training cross-entropy plus $0.01\|Y\|_{\mathrm{F}}^2/2$, and the UL objective is cross-entropy on a separate validation set. The LL-training, UL-validation, and test sample sizes are 256, 128, and 128, respectively.

The teacher representation has rank 8. Inputs are generated as $z = \xi U \,\mathrm{diag}(s) U^\top$, where ξ is standard Gaussian, U is a fixed orthogonal matrix, and the entries of s are geometrically spaced with ratio κ_{in} and normalized such that $\sum_i s_i^2 = 16$. Thus, the population input-covariance condition number is κ_{in}^2 while the total input variance is fixed. Labels are generated by the same low-rank tanh teacher construction for each value of κ_{in}.

Methods and hyperparameters. All five methods use the same warm-started stochastic bilevel estimator: $D = 5$, $Q = 4$, $\alpha = 0.1$, $\eta = 0.2$, $B = 64$, $\theta = 0.9$. NS-BiMuon uses the cubic Taylor iteration with degree $d = 1$, depth $q = 12$, and initialization $\frac{M}{\max\{1, \|M\|_{\mathrm{F}}\}}$. All methods use

the EMA momentum in (5) and differ only in the final UL transformation. OpNorm and FrobNorm use denominator floor 10^{-30}. Exact-polar BiMuon applies the supported polar factor to the same stochastic hypergradient momentum.

Tuning and evaluation. Hyperparameters are selected using discovery seeds 0 and 1. The selected representative setting uses $\kappa_{\text{in}} = 10$. All methods use the same 17-point UL-stepsize candidate grid, with a separate stepsize selected for each method. The grid is

$$\beta \in \{0.00625,\ 0.0125,\ 0.025,\ 0.05,\ 0.1,\ 0.2,\ 0.4,\ 0.8,\ 1.6,\ 3.2,\ 6.4,\ 12.8,\ 25.6,\ 51.2,\ 102.4,\ 204.8,\ 409.6\}.$$

The resulting solver parameters and stepsizes are frozen before evaluation on seeds 10–14. All evaluation runs use 60 UL iterations without retuning or early stopping.

At each logged X, the LL classifier is independently refit using L-BFGS, and the primary metric is the resulting optimized UL validation cross-entropy. Curves report means and standard errors over the five evaluation seeds.

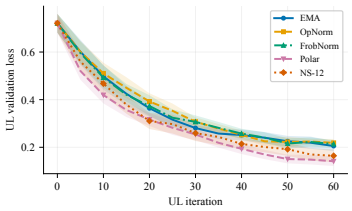


Figure 5: Synthetic nonlinear representation learning at $\kappa_{\text{in}} = 10$. Curves show mean optimized UL validation loss and standard-error bands over five evaluation seeds.

Results. At $\kappa_{\text{in}} = 10$, exact-polar BiMuon achieves final optimized UL validation loss 0.1414 ± 0.0452, NS-BiMuon achieves 0.1643 ± 0.0602, and stocBiO+EMA achieves 0.2054 ± 0.0469, where uncertainties are sample standard deviations over the five evaluation seeds. The two BiMuon variants therefore provide an additional controlled comparison of spectral reshaping against the same scalar-normalized baselines used in the real-data experiment.

We also vary the input anisotropy while keeping the representation rank, solver depths, NS normalization, and stopping rule fixed. Figure 6 reports the selected $\kappa_{\text{in}} = 10$ configuration together with the anisotropy sweep.

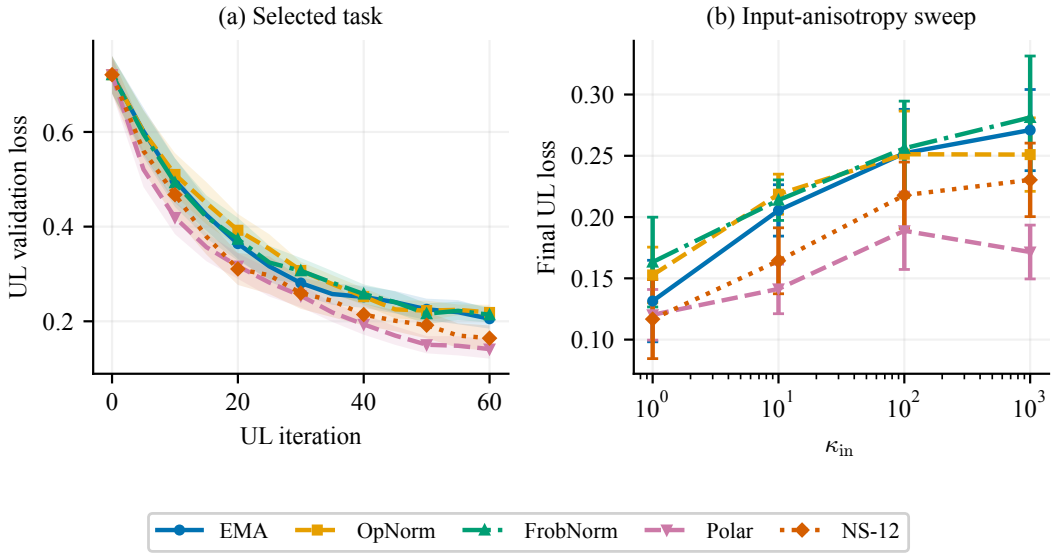


Figure 6: Supplementary nonlinear representation-learning results. Left: the frozen configuration at $\kappa_{\text{in}} = 10$. Right: the input-anisotropy sweep, with method-specific UL stepsizes selected on discovery seeds at each value. Bands and error bars show standard errors over five evaluation seeds. The optimized UL validation loss is computed after independently refitting the LL classifier.

F.4 Supplementary Controlled Blockwise Allocation

Setup. We further study the blockwise allocation rule on a controlled quadratic bilevel problem with $J = 4$ UL blocks $X^{(b)} \in \mathbb{R}^{16\times 16}$ and an LL matrix $Y \in \mathbb{R}^{64\times 16}$. Let $P_b \in \mathbb{R}^{64\times 16}$ embed

block b into its corresponding 16 rows, so that $P_b^\top P_c = 0\,(b \neq c)$, $P_b^\top P_b = I$. We use $g(X,Y) = \frac{1}{2}\langle Y, HY\rangle - \left\langle \sum_{b=1}^4 c_b P_b X^{(b)}, Y \right\rangle$, and $f(X,Y) = \frac{1}{2}\|CY - R\|_{\mathrm{F}}^2$.

The LL curvature matrix is $H = U\,\mathrm{diag}(h)U^\top$, where U is obtained by QR factorization of a standard Gaussian matrix and the 64 eigenvalues are logarithmically spaced from 1 to 10. Thus $\mu = 1$, $\kappa(H) = 10$. The dense matrix $C \in \mathbb{R}^{64\times 64}$ has independent Gaussian-QR left and right orthogonal factors and singular values linearly spaced from 0.5 to 1.5. We set $R = CY_{\mathrm{target}}$, where the entries of Y_{target} are independently drawn from $\mathcal{N}(0, 1/64)$.

The exact LL solution is $Y^\star(X) = H^{-1}\sum_{b=1}^4 c_b P_b X^{(b)}$. We set $c_b = \rho^{(b-1)/3}$, so that ρ is the block-sensitivity ratio. The orthogonal embeddings give $\|\nabla_Y g(X,Y) - \nabla_Y g(X',Y)\|_{\mathrm{F}}^2 = \sum_{b=1}^4 c_b^2 \|X^{(b)} - X'^{(b)}\|_{\mathrm{F}}^2$. Hence $L_{XY,b} = c_b$, $r_b = 16$, $w_b = 16c_b^2$. For each seed, H, C, and R are fixed across all allocation rules and all values of ρ. All runs start from $X_0 = 0$, $Y_0 = 0$, $M_{-1} = 0$.

Methods and hyperparameters. We use the deterministic full-gradient specialization of the same bilevel recurrence, with $D = 5$, $\alpha = 0.1$, $Q = 4$, $\eta = 0.1$, $\theta = 0.9$. For each block, $O_k^{(b)} = \mathrm{polar}(M_k^{(b)})$, $a_{b,k} = \langle M_k^{(b)}, O_k^{(b)}\rangle$. The supported polar implementation retains singular values larger than 10^{-14} times the largest singular value and maps zero momentum to zero. All runs use 100 UL iterations and CPU float64 arithmetic.

The raw allocation weights are

$$\widetilde{\omega}_{b,k} = \begin{cases} 1, & \text{Uniform,} \\ a_{b,k}, & \text{Progress-only,} \\ 1/w_b, & \text{Sensitivity-only,} \\ a_{b,k}/w_b, & \text{Proposed.} \end{cases}$$

Every rule is normalized as $\omega_{b,k} = \frac{\widetilde{\omega}_{b,k}}{\sqrt{\sum_c w_c \widetilde{\omega}_{c,k}^2}}$, followed by $X_{k+1}^{(b)} = X_k^{(b)} - \beta\omega_{b,k}O_k^{(b)}$. Positive raw weights are divided by their maximum before normalization for numerical stability. If all progress scores vanish, the update is set to zero; otherwise $\sum_b w_b\omega_{b,k}^2 = 1$.

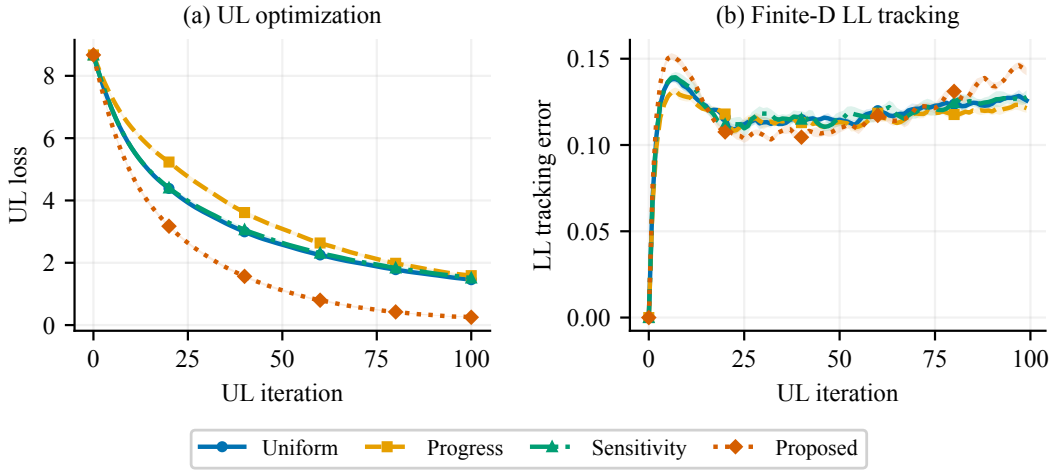


Figure 7: Controlled blockwise relative stepsize allocation at $\rho = 10$. All methods use the same global stepsize and theoretical LL-drift budget; only the relative allocation across matrix blocks differs. Curves show means and standard-error bands over five evaluation seeds.

Tuning and evaluation. The pilot uses seeds 0 and 1, $\rho \in \{1, 10, 30\}$, $\beta \in \{0.03, 0.1, 0.3, 1.0\}$, for all four allocation rules. One common β is selected by minimizing the mean final-to-initial UL objective ratio over the pilot runs, with ties resolved in favor of the smaller stepsize. This selects $\beta = 0.3$. The value is frozen for evaluation on seeds 100–104 and $\rho \in \{1, 3, 10, 30\}$.

The optimized UL objective is evaluated using the exact LL solution: $\Phi(X_k) = f(X_k, Y^\star(X_k))$. We additionally report $\|Y_k^D - Y^\star(X_k)\|_{\mathrm{F}}$ as the finite-D LL tracking error. Curves and error bars

report means and standard errors over the five paired evaluation seeds; table entries report sample standard deviations.

Results. Figure 7 reports the representative $\rho = 10$ experiment under the common global stepsize $\beta = 0.3$. The proposed allocation achieves lower UL objective values than Uniform, Progress-only, and Sensitivity-only under the same blockwise drift budget.

Table 3: Final optimized UL objective $\Phi(X_{100})$ under the common $\beta = 0.3$. Entries are means and sample standard deviations over five evaluation seeds.

ρ	Uniform	Progress-only	Sensitivity-only	Proposed
1	$.2536 \pm .0258$	$.2512 \pm .0212$	$.2536 \pm .0258$	$.2512 \pm .0212$
3	$.5217 \pm .0408$	$.4753 \pm .0405$	$.5750 \pm .0868$	$.2512 \pm .0212$
10	$1.4566 \pm .0711$	$1.5870 \pm .0719$	$1.5110 \pm .1253$	$.2512 \pm .0212$
30	$2.4444 \pm .1239$	$2.8098 \pm .1317$	$2.5259 \pm .1470$	$.2512 \pm .0212$

Table 3 and Figure 8 show the complete sensitivity-ratio sweep. For $\rho \in \{3, 10, 30\}$, the proposed allocation attains a lower final UL objective than all three alternatives on all five paired evaluation seeds. At $\rho = 1$, Uniform and Sensitivity-only coincide, while Progress-only and Proposed coincide.

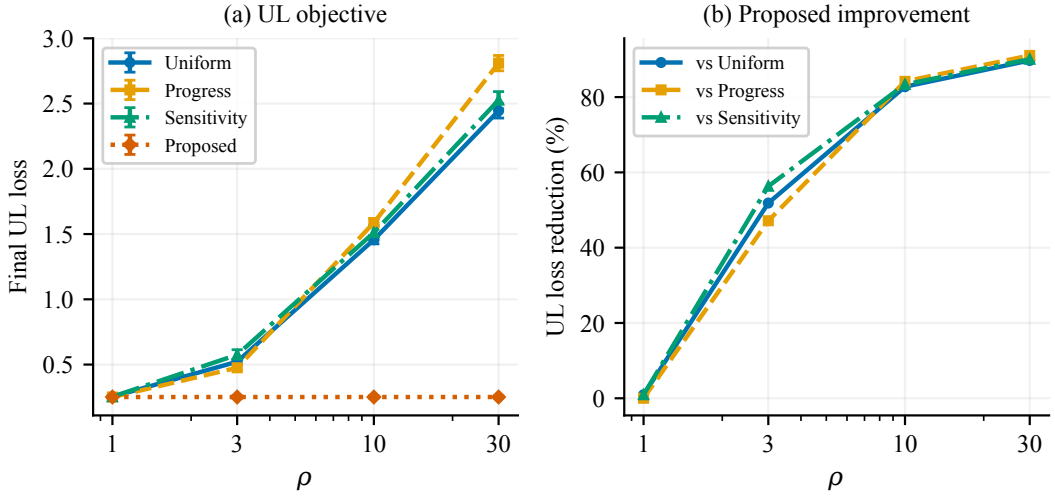


Figure 8: Sensitivity-ratio sweep for controlled blockwise allocation with the common $\beta = 0.3$. Means and standard errors are computed over five evaluation seeds. The same problem matrices and initialization are reused across ρ within each seed.

Scale invariance. The constant proposed trajectory across ρ follows from the scaling structure of this construction. Writing $Z^{(b)} = c_b X^{(b)}$ and $M_k^{(b)} = c_b N_k^{(b)}$ gives $a_{b,k} = c_b \widetilde{a}_{b,k}$ and $w_b = 16 c_b^2$. Positive scalar multiplication does not change the polar direction, and the factor c_b cancels from the effective proposed update in $Z^{(b)}$. Therefore, under the shared zero initialization, the proposed (Z, Y) trajectories are invariant to ρ.