BROS: Bias-Corrected Randomized Subspaces for Memory-Efficient Single-Loop Bilevel Optimization

Hengrui Zhang[*]
Peking University
zhanghengrui26@stu.pku.edu.cn

Boao Kong[*]
Peking University
kongboao@stu.pku.edu.cn

Engao Zhang[*]
Peking University
2300012403@stu.pku.edu.cn

Kun Yuan[†]
Peking University
kunyuan@pku.edu.cn

Abstract

Stochastic bilevel optimization (SBO) has become a standard framework for hyperparameter learning, data reweighting, representation learning, and data-mixture optimization in deep learning. Existing exact single-loop SBO methods and memory-efficient surrogate SBO methods either create severe memory pressure for large lower-level neural networks or lack competitive convergence guarantees under standard assumptions. In this paper, we propose **BROS**, a memory-efficient single-loop SBO method with the same convergence rate order as exact single-loop SBO methods. **BROS** performs lower and auxiliary updates in randomized subspaces with a Rademacher bi-probe correction that recovers an unbiased Hessian-action estimator. We prove that **BROS** preserves the $\mathcal{O}(\varepsilon^{-2})$ sample complexity of MA-SOBA for finding an ε-stationary point under only standard assumptions. Experiments on hyper-data cleaning, data-mixture learning, hyper-representation learning, and ViT sample reweighting show that **BROS** reduces peak memory by up to 44.9% while closely matching full-space baseline performance.

1 Introduction

Stochastic bilevel optimization (SBO) has become an increasingly important framework in deep learning, with applications including hyper-cleaning and noisy-label reweighting [20, 61, 54, 68], hyperparameter optimization and meta-learning [13, 48, 7], hyper-representation learning [27, 65], and data reweighting or data-selection for large-scale training [12, 58, 50, 45, 56]. Across these applications, the upper-level variable can take a variety of forms. The lower-level problem, however, typically involves training a neural network with multilayer matrix parameters. Based on this observation, this paper studies a more general multilayer matrix-parameterized SBO model, which covers the vector formulation used in previous works [16, 10, 7, 9]:

$$\min_{x\in\mathbb{R}^{d_x}} \ \Phi(x) := f\big(x, \boldsymbol{Y}^*(x)\big), \quad \boldsymbol{Y}^*(x) \in \operatorname{argmin}_{\boldsymbol{Y}\in\mathcal{Y}} g(x,\boldsymbol{Y}), \tag{1a}$$

$$\text{where } \ \mathcal{Y} := \Big\{\boldsymbol{Y} = (Y_1,\dots,Y_L) \,:\, Y_\ell \in \mathbb{R}^{m_\ell\times n_\ell},\ \ell\in[L]\Big\}. \tag{1b}$$

Here x denotes the upper-level variable, and $\boldsymbol{Y}$ collects the trainable parameters of the lower-level neural network. We work in the stochastic setting where the objectives are population risks:

$$f(x,\boldsymbol{Y}) = \mathbb{E}_{\omega_f\sim\mathcal{D}_f}[\ell_f(x,\boldsymbol{Y};\omega_f)], \quad g(x,\boldsymbol{Y}) = \mathbb{E}_{\omega_g\sim\mathcal{D}_g}[\ell_g(x,\boldsymbol{Y};\omega_g)],$$

[*]Equal contribution.
[†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

and the algorithm accesses these objectives through stochastic oracles.

Full-space hypergradient methods. A large body of algorithms has been developed for SBO. A common route is to estimate the hypergradient $\nabla\Phi(x)$ either by approximating the inverse Hessian or by maintaining an auxiliary linear-system variable updated through Hessian-vector products (HVPs) and Jacobian-vector products (JVPs) [16, 29]. This route includes early stochastic bilevel methods [16, 24, 28] and recent single-loop methods such as SOBA/SABA [10] and MA-SOBA [7], which serve as the main baselines for our theoretical convergence rate comparison. Other variance-reduced or momentum-based baselines such as SPABA [9], SUSTAIN [29], and MRBO [59] improve sample complexity but require finite-sum structure or stronger oracle assumptions, so they are not the primary convergence-rate baselines for our comparison here. These methods provide a clean hypergradient baseline, but face a severe limitation:

> **Limitation 1.** *Full-space hypergradient methods remain memory-heavy for large neural networks.*

Although these methods need not explicitly form the full Hessian or Jacobian, their recursions still rely on full-space second-order information and auxiliary linear-system variables. When the lower-level variable is a large neural network with hundreds of millions or billions of parameters, the corresponding Hessian and Jacobian matrices are impossible to form or store, and even HVP-based full-space auxiliary recursions can impose prohibitive memory and computational costs.

Memory-efficient surrogate methods. Another way to reduce the cost is to move away from the standard hypergradient update. Penalty methods [42, 34] replace the original bilevel problem with a penalty reformulation. Value-function-based fully first-order methods [35] avoid HVP and JVP. FdeHBO [63] uses finite differences to avoid explicit HVP and JVP. Zeroth-order methods [51] further reduce the need for full derivative information. These approaches can substantially reduce or avoid explicit Hessian/Jacobian-vector computations, but this scalability typically comes with one of three costs: no exact recovery of the original bilevel solution, slower stochastic complexity, or additional structural/regularity assumptions:

> **Limitation 2.** *Surrogate methods theoretically lose the exact hypergradient baseline update.*

Penalty methods optimize a finite-penalty surrogate rather than the original problem, with exact recovery requiring a limiting penalty regime and additional regularity conditions [42, 34]. Value-function-based fully first-order methods typically have worse stochastic complexity orders than standard single-loop hypergradient methods [35]. Finite-difference methods such as FdeHBO [63] rely on an auxiliary projection and higher-order smoothness assumptions to control finite-difference errors. Zeroth-order bilevel methods often require many more function evaluations and stronger structural assumptions, such as low effective rank [51]. Thus, these approaches do not answer the question we study here: *can one reduce memory usage while preserving the standard single-loop hypergradient update and its $\mathcal{O}(\varepsilon^{-2})$ sample complexity for finding an ε-stationary point?*

Randomized-subspace large-model training. Recent progress in single-level large-model training suggests a complementary path. Several studies have observed and exploited low-rank or low-dimensional structure in large-model gradients and updates, leading to memory-efficient methods that store and optimize only subspace variables. For example, LoRA [26] constrains trainable updates to low-rank adapters, GaLore [67] projects gradient matrices into low-rank subspaces to reduce optimizer-state memory, and RSO [8] decomposes LLM training into a sequence of lower-dimensional subproblems. Importantly for our setting, RSO shows that randomized subspace training can reduce activation and gradient memory while retaining the same convergence order as full-space training in single-level optimization [8]. Motivated by this, we ask whether the lower-level and auxiliary-level training problem in SBO can also be solved in randomized subspaces.

Unfortunately, applying randomized subspace methods to SBO is highly nontrivial: unlike first-order RSO in single-level optimization, exact bilevel methods must compress and recover second-order information for the hypergradient. One may think of using an RSN-like strategy [17]: compute Hessian information inside the sampled subspace and lift it back to the original full space. This is sufficient for randomized Newton methods, whose analysis treats the sketched Newton step itself as a descent direction. However, in exact bilevel optimization, the Hessian determines the auxiliary linear-system used in the implicit-gradient formula. A biased lifted Hessian makes the auxiliary variable track the wrong system and biases the hypergradient.

Table 1: Rate- and assumption-oriented comparison of representative stochastic bilevel algorithms. N_ε denotes the oracle/iteration complexity for obtaining $\mathbb{E}\|\nabla\Phi(\bar{x})\|^2 \le \varepsilon$, following each paper's own oracle model and stationarity convention.

Algorithm	S.L.[†]	Oracle[‡]	Assmp.[▷]	E.Assmp.[◁]	N_ε	Std. ε^{-2}?[♦]	U.S.[♡]
BSA [16]	×	H/JVP	Std.	–	$\widetilde{\mathcal{O}}(\varepsilon^{-3})$	×	Full
SOBA [10]	✓	H/JVP	Str.	HO	$\mathcal{O}(\varepsilon^{-2})$	HO only	Full
SUSTAIN/MRBO [29, 59]	✓	H/JVP	Str.	MS	$\widetilde{\mathcal{O}}(\varepsilon^{-3/2})$	MS only	Full
F^{2}SA [35]	✓	FO	Diff.	FO	$\widetilde{\mathcal{O}}(\varepsilon^{-7/2})$	×	Full
F^{3}SA [35]	✓	FO	Diff.	FO	$\widetilde{\mathcal{O}}(\varepsilon^{-5/2})$	×	Full
FdeHBO [63]	✓	FO	Diff.	FD+Proj.	$\widetilde{\mathcal{O}}(\varepsilon^{-3/2})$	Diff.	Full
SPABA-E [9]	✓	H/JVP	Str.	MS	$\mathcal{O}(\varepsilon^{-3/2})$	MS only	Full
SPABA-FS [9]	✓	H/JVP	Std.	–	$\mathcal{O}((n+m)^{1/2}\varepsilon^{-1})$	FS	Full
BROS	✓	H/JVP	**Std.**	RP	$\mathcal{O}(\varepsilon^{-\mathbf{2}})$	✓	**Proj.**
MA-SOBA [7]	✓	H/JVP	Std.	–	$\mathcal{O}(\varepsilon^{-2})$	✓	Full

[†] The algorithm is a single-loop method.

[‡] Derivative oracle. H/JVP = Hessian-/Jacobian-vector products, FO = first-order-only.

[▷] Coarse assumption regime for comparability with the standard SBO baseline. Std. = standard nonconvex–strongly-convex stochastic bilevel setting with smooth f, strongly convex and second-order smooth g, and unbiased bounded-variance stochastic gradient/HVP/JVP oracles; Str. = stronger assumptions than Std.; Diff. = different oracle model or reformulation.

[◁] Specific extra condition, oracle restriction, or mechanism explaining the entry in 'Assmp.'. HO = high-order smoothness, MS = mean-squared smoothness of stochastic oracles, FD+Proj. = finite-difference approximation with auxiliary projection, RP = randomized-projection moment conditions with no block-diagonal lower-Hessian assumption.

[♦] The method attains the standard $\mathcal{O}(\varepsilon^{-2})$ complexity under Std. assumption. "HO only", "MS only", "Diff.", and "FS" mean that the displayed rate requires, respectively, additional high-order smoothness, mean-squared smoothness of stochastic oracles, a different oracle model/reformulation, or a finite-sum setting.

[♡] Lower/Auxiliary update space. Full = full trainable space, Proj. = projected/randomized subspace.

To overcome this, we propose **BROS** (**B**ilevel optimization via **R**and**O**mized **S**ubspace), a memory-efficient single-loop stochastic bilevel method for the multilayer matrix-parameterized problem (1). **BROS** introduces a Rademacher-probe-based correction that recovers an unbiased hypergradient estimator while retaining the memory advantage of randomized subspace training and the convergence rate order of exact single-loop methods. The core contributions are summarized as follows:

C1. Memory-efficient randomized-subspace bilevel recursion. We develop a single-loop stochastic bilevel method that decomposes the lower-level neural-network training recursion into randomized lower-dimensional matrix subproblems. Since each subproblem operates in a reduced subspace, **BROS** lowers the dimensionality and memory costs of the lower gradients, auxiliary directions, activations, and HVP/JVP computations required by the lower-level recursion.

C2. Unbiased hypergradient recovery with MA-SOBA-order convergence. We identify the projection-induced bias in the lifted auxiliary Hessian action and remove it using a layerwise Rademacher bi-probe correction. The resulting estimator is conditionally unbiased for the full multilayer Hessian action without imposing additional structural assumptions. Under standard assumptions in SBO, this correction allows **BROS** to match the $\mathcal{O}(\varepsilon^{-2})$ sample complexity of the MA-SOBA backbone for finding an ε-stationary point.

C3. Empirical validation on neural and language-model bilevel tasks. We evaluate **BROS** on MNIST hyper-data cleaning, data-mixture learning with 280M GPT-style proxy/main models, CIFAR-10 hyper-representation learning, and CIFAR-100 ViT sample reweighting. **BROS** closely matches full-space MA-SOBA on hyper-data cleaning and data-mixture learning, while reducing proxy-training peak memory by about 45% relative to MA-SOBA and about 27% relative to Penalty.

Table 1 summarizes the positioning of **BROS** by separating the oracle route, assumption regime, convergence order, and whether the lower-level and auxiliary computations are full-space. A detailed discussion of related work is provided in Appendix A.

Notations. For an integer L, let $[L] := \{1, \dots, L\}$. All vector norms are Euclidean norms, and all matrix norms are Frobenius norms unless otherwise specified. For random variables, $\mathbb{E}[\cdot]$ and $\mathbb{E}[\cdot \mid \mathcal{F}]$ denote expectation and conditional expectation, and $\mathcal{F}_k$ denotes the filtration before drawing

the projector at iteration k. We use bold uppercase letters, such as $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{Y}, \boldsymbol{Z}$, for multilayer matrix variables with the same layerwise structure as $\boldsymbol{Y} = (Y_1, \ldots, Y_L)$.

The ℓ-th block of $\boldsymbol{U}$ is denoted by U_ℓ. For two block variables with compatible dimensions, $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ denote the product Frobenius inner product and norm. For the lower-level objective g, write $\mathcal{H}(x, \boldsymbol{Y}) := \nabla^2_{\boldsymbol{YY}} g(x, \boldsymbol{Y})$ and $\mathcal{J}(x, \boldsymbol{Y}) := \nabla^2_{x\boldsymbol{Y}} g(x, \boldsymbol{Y})$. For a block linear operator $\mathcal{A} : \mathcal{Y} \to \mathcal{Y}$, $\mathcal{A}_{\ell t}$ denotes its (ℓ, t)-block operator from block t to block ℓ. At iteration k, we write $\mathcal{H}^k := \mathcal{H}(x^k, \boldsymbol{Y}^k)$ and $\mathcal{H}^k_{\ell t}$ for the (ℓ, t)-block of $\mathcal{H}^k$. When multilayer projectors appear, $P_\ell \in \mathbb{R}^{m_\ell \times r_\ell}$ with $r_\ell \le m_\ell$, and products such as $\boldsymbol{PB}$ and $\boldsymbol{P}^\top \boldsymbol{U}$ are understood blockwise.

2 Stochastic bilevel optimization via randomized subspace (BROS)

This section develops **BROS** through a simplified single-layer case with $L = 1$, i.e., the case where the lower-level variable Y is a single-layer matrix in $\mathbb{R}^{m \times n}$. We make this simplification only for clarity of exposition, the canonical multi-layer derivation is deferred to Appendix B.1.

2.1 Hypergradient computation and single-loop bilevel updates

When solving the upper-level problem, the core is the hypergradient $\nabla \Phi(x)$ [16]. Under Assumption 1, the lower-level solution $Y^*(x)$ is unique, and the implicit-gradient formula gives:

$$\nabla \Phi(x) = \nabla_x f(x, Y^*(x)) - \mathcal{J}(x, Y^*(x)) \big[\mathcal{H}(x, Y^*(x))^{-1} [\nabla_Y f(x, Y^*(x))] \big] . \tag{2}$$

Evaluating this expression directly is expensive because it involves the inverse of the lower Hessian operator. Following the standard exact-hypergradient route, we introduce an auxiliary variable

$$Z^*(x) := \mathcal{H}(x, Y^*(x))^{-1} [\nabla_Y f(x, Y^*(x))].$$

Equivalently, $Z^*(x)$ is the solution to the quadratic auxiliary problem

$$Z^*(x) = \operatorname{argmin}_{Z \in \mathbb{R}^{m \times n}} \left\{ \frac{1}{2} \langle Z, \mathcal{H}(x, Y^*(x))[Z] \rangle - \langle Z, \nabla_Y f(x, Y^*(x)) \rangle \right\}. \tag{3}$$

Once $Z^*(x)$ is obtained, the hypergradient reduces to $\nabla_x f(x, Y^*(x)) - \mathcal{J}(x, Y^*(x))[Z^*(x)]$.

Following this idea, solving the bilevel problem essentially involves three coupled subproblems. Define $h(x, Y, Z) := \frac{1}{2} \langle Z, \mathcal{H}(x, Y)[Z] \rangle - \langle Z, \nabla_Y f(x, Y) \rangle$. Then

$$x^* \in \operatorname{argmin}_{x \in \mathbb{R}^{d_x}} f(x, Y^*(x)), \qquad \text{(upper-level)} \tag{4a}$$

$$Y^*(x) = \operatorname{argmin}_{Y \in \mathbb{R}^{m \times n}} g(x, Y), \qquad \text{(lower-level)} \tag{4b}$$

$$Z^*(x) = \operatorname{argmin}_{Z \in \mathbb{R}^{m \times n}} h(x, Y^*(x), Z). \qquad \text{(auxiliary-level)} \tag{4c}$$

Given x, the exact-hypergradient procedure first obtains $Y^*(x)$ by solving the lower-level problem, then solves the auxiliary-level problem with $Y^*(x)$ fixed to obtain $Z^*(x)$, and finally uses $Z^*(x)$ to form the hypergradient for the upper-level update. The auxiliary formulation in (4) therefore leads to a three-level computational template. Rather than solving the lower-level and auxiliary-level problems to high accuracy in nested loops, recent exact single-loop methods perform one stochastic update for each level per iteration, yielding a coupled single-loop recursion over (x, Y, Z) [10, 7].

Memory bottleneck in large-scale lower-level problems. In the full-space single-loop template, the lower update uses a stochastic lower gradient, the auxiliary update uses HVPs to solve the quadratic system for Z, and the upper update substitutes the auxiliary variable into the hypergradient formula. This creates a substantial memory bottleneck. For a lower-level block $Y \in \mathbb{R}^{m \times n}$, the full-space recursion must maintain the lower variable Y^k and the auxiliary variable Z^k, and the derivative evaluations introduce additional $O(mn)$ memory for lower gradients, auxiliary residuals, HVP inputs and outputs, and JVP-related quantities. Moreover, HVP/JVP computations are implemented through higher-order automatic differentiation, which can retain multiple forward/backward tapes and temporary tensors during one iteration. Thus, the peak trainable-side memory can be several times larger than the memory needed to store the lower model itself, creating a large overhead for high-dimensional neural networks and potentially translating into substantially higher accelerator-memory requirements and hardware provisioning costs.

2.2 Randomized subspace sketching for bilevel recursions

To reduce the memory overhead described above, a natural idea is to combine randomized subspace optimization (RSO) [23, 8] for the lower-level update with randomized subspace Newton (RSN) sketching [17] for the auxiliary update.

Specifically, at iteration k, we sample a scaled Haar projector $P^k \in \mathbb{R}^{m\times r}$, i.e., $P^k = \sqrt{m/r}\, O_r^k$ where O_r^k contains the first r columns of a Haar-uniform orthogonal matrix. This uniform distribution is invariant under orthogonal transformations. This normalization gives $(P^k)^\top P^k = (m/r)I_r$ and $\mathbb{E}[P^k(P^k)^\top \mid \mathcal{F}_k] = I_m$. We then search for the update to Y^k inside the subspace induced by P^k:

$$B^k = \operatorname{argmin}_{B\in\mathbb{R}^{r\times n}} \tilde{g}_k(x^k, B), \quad \tilde{g}_k(x, B) := g(x, Y^k + P^k B). \tag{5}$$

After solving this reduced problem, the full-space lower variable is updated by $Y^{k+1} = Y^k + P^k B^k$.

For the auxiliary variable, the analogous sketching step computes Hessian and Jacobian actions only with respect to the reduced variable B. For $W \in \mathbb{R}^{r\times n}$, define the projected HVP and JVP by

$$\tilde{\mathcal{H}}^{k+1}[W] := \nabla^2_{BB}\tilde{g}_k(x^k, 0)[W], \quad \tilde{\mathcal{J}}^{k+1}[W] := \nabla^2_{xB}\tilde{g}_k(x^k, 0)[W]. \tag{6}$$

The vanilla sketch-then-lift auxiliary update would then replace the full-space HVP and JVP by

$$\widehat{\mathcal{H}}^{k+1}_{\text{lift}}[Z^k] := P^k\tilde{\mathcal{H}}^{k+1}[(P^k)^\top Z^k], \quad \widehat{\mathcal{J}}^{k+1}_{\text{lift}}[Z^k] := \tilde{\mathcal{J}}^{k+1}[(P^k)^\top Z^k]. \tag{7}$$

This construction replaces the $O(mn)$ full-space HVP/JVP directions and their associated temporary memory with projected $O(rn)$ derivative directions. Thus, before the final lifting step, the lower-gradient, HVP, and JVP computations are carried out in the randomized subspace, directly reducing the memory overhead identified in Section 2.1.

Key limitation for vanilla sketch-then-lift. The key limitation is that the vanilla sketch-then-lift estimator does not, in general, recover the full-space HVP required by the exact hypergradient formula. First, the lifted Hessian action in (7) is generally a biased estimate of the full-space Hessian action: even when P^k is sampled from a Haar distribution and satisfies $\mathbb{E}[P^k(P^k)^\top \mid \mathcal{F}_k] = I_m$, there is *no* guarantee for $\mathbb{E}[\widehat{\mathcal{H}}^{k+1}_{\text{lift}}[Z^k] \mid \mathcal{F}_k] = \mathcal{H}^k[Z^k]$. The auxiliary recursion therefore tracks an inexact linear system, which distorts the hypergradient and can lead to a nonstationary limit. Appendix C.5 gives a deterministic quadratic counterexample for this issue. Second, any recovery mechanism for the above-mentioned distortion must preserve the subspace memory advantage. In particular, it *cannot* rely on forming full Hessian/Jacobian matrices, computing full-space HVPs/JVPs, or incurring full-space derivative memory.

2.3 Rademacher-probe recovery of auxiliary HVPs

The bias arises because the same sketch acts on both sides of each Hessian block. With $\tilde{Z}^k := (P^k)^\top Z^k$ and $A^{k+1} := P^k\tilde{\mathcal{H}}^{k+1}[\tilde{Z}^k]$, the Haar fourth-moment identity yields

$$\mathbb{E}[(A^{k+1})_{:,i} \mid \mathcal{F}_k] = \sum_j \big[(a-b)H^k_{ij} + b(H^k_{ij})^\top + b\,\mathrm{tr}(H^k_{ij})I_m\big] z^k_j, \tag{8}$$

where $Z = [z_1, \dots, z_n]$, $(\mathcal{H}^k[Z])_{:,i} = \sum_j H^k_{ij} z_j$, and a, b are the single-layer coefficients in (29).

To remove the extra transpose and trace terms, **BROS** samples independent Rademacher probes $u^{k+1} \in \{-1, 1\}^n$ and $\xi^{k+1} \in \{-1, 1\}^r$, and uses one additional subspace HVP $M^{k+1} := \tilde{\mathcal{H}}^{k+1}[\xi^{k+1}(u^{k+1})^\top]$ to form

$$\widehat{C}^{k+1} := Z^k\Big((M^{k+1})^\top\xi^{k+1}(u^{k+1})^\top\Big), \quad \widehat{B}^{k+1}_\sharp := P^k\Big(\xi^{k+1}\big((M^{k+1})^\top(\tilde{Z}^k u^{k+1})\big)^\top\Big). \tag{9}$$

For Haar projectors [8] P^k with $r \geq 2$, combining these terms gives an unbiased full-space HVP estimator as follows:

$$\widehat{\mathcal{H}[Z]}^{k+1} := \frac{r(m-1)(m+2)}{m(mr+m-2)}A^{k+1} - \frac{m-r}{mr+m-2}\widehat{C}^{k+1} + \frac{r(m-1)(m-r)}{m(r-1)(mr+m-2)}\big(A^{k+1} - \widehat{B}^{k+1}_\sharp\big). \tag{10}$$

For later use, we denote the corrected estimator in (9)–(10) by

$$\mathrm{CorrHVP}(P^k, Z^k, \tilde{\mathcal{H}}^{k+1}; u^{k+1}, \xi^{k+1}) := \widehat{\mathcal{H}[Z]}^{k+1}.$$

Algorithm 1 BROS

Require: stepsizes $\{\alpha_k, \beta_k, \gamma_k, \theta_k\}$, initial $(x^0, \boldsymbol{Y}^0, \boldsymbol{Z}^0, h^0 = 0)$

for $k = 0, 1, \ldots, K-1$ **do**
  Sample projectors $\boldsymbol{P}^k$ and Rademacher probes $\{u_\ell^{k+1}, \xi_\ell^{k+1}\}_{\ell=1}^L$
  Obtain projected stochastic oracles $(\tilde{U}_x^{k+1}, \tilde{\boldsymbol{U}}_Y^{k+1}, \tilde{\boldsymbol{V}}^{k+1}, \tilde{\mathcal{H}}^{k+1}, \tilde{\mathcal{J}}^{k+1})$
  for $\ell = 1, \ldots, L$ **do**
    $\widehat{\mathcal{H}_{\ell\ell}[Z_\ell]}^{k+1} \leftarrow \mathrm{CorrHVP}_\ell\Big(P_\ell^k, Z_\ell^k, \tilde{\mathcal{H}}^{k+1}; u_\ell^{k+1}, \xi_\ell^{k+1}\Big)$ ▷ diagonal correction by (10)
    $(\widehat{\mathcal{H}[\boldsymbol{Z}]}^{k+1})_\ell \leftarrow \widehat{\mathcal{H}_{\ell\ell}[Z_\ell]}^{k+1} + \sum_{t \neq \ell} P_\ell^k \big(\tilde{\mathcal{H}}^{k+1}[E_t((P_t^k)^\top Z_t^k)]\big)_\ell$ ▷ off-diagonal blocks lifting
  end for
  $x^{k+1} \leftarrow x^k - \alpha_k h^k$ ▷ upper-level update
  $\boldsymbol{Y}^{k+1} \leftarrow \boldsymbol{Y}^k - \beta_k \boldsymbol{P}^k \tilde{\boldsymbol{V}}^{k+1}$ ▷ lower-level update
  $\boldsymbol{Z}^{k+1} \leftarrow \boldsymbol{Z}^k - \gamma_k \big(\widehat{\mathcal{H}[\boldsymbol{Z}]}^{k+1} - \boldsymbol{P}^k \tilde{\boldsymbol{U}}_Y^{k+1}\big)$ ▷ auxiliary-level update
  $h^{k+1} \leftarrow (1-\theta_k) h^k + \theta_k \big(\tilde{U}_x^{k+1} - \tilde{\mathcal{J}}^{k+1}[\boldsymbol{P}^{k\top} \boldsymbol{Z}^k]\big)$ ▷ moving average update
end for

This subroutine provides the debiased auxiliary Hessian-action direction used in place of the vanilla lifted HVP in the Z-update.

For the multilayer setting, **BROS** applies this correction layerwise to the diagonal Hessian blocks and directly lifts the off-diagonal projected HVPs. The detailed blockwise construction process is deferred to Appendix B.1. In Algorithm 1, $E_t(W)$ inserts W into the t-th projected block and zeros elsewhere, and $\mathrm{CorrHVP}_\ell$ denotes the layer-ℓ version of CorrHVP with layerwise dimensions, projector, and probes. Algorithm 1 presents the resulting multilayer single-loop recursion: after constructing the corrected auxiliary HVP estimator, it performs one coupled update for the upper-level variable, lower-level variable, auxiliary variable, and the moving-average hypergradient estimator h^k. The moving-average recursion for h^k is a standard variance-reduction device in single-loop SBO algorithms, used to stabilize stochastic hypergradient evaluation [7, 32].

Algorithm 1 takes one stochastic step from $\boldsymbol{B} = \boldsymbol{0}$ instead of solving (5) exactly. Here $(\tilde{U}_x^{k+1}, \tilde{\boldsymbol{U}}_Y^{k+1}, \tilde{\boldsymbol{V}}^{k+1})$ estimate $(\nabla_x \tilde{f}_k, \nabla_{\boldsymbol{B}} \tilde{f}_k, \nabla_{\boldsymbol{B}} \tilde{g}_k)$ at $(x^k, \boldsymbol{0})$, with $\tilde{f}_k$ defined analogously to $\tilde{g}_k$; $\tilde{\mathcal{H}}^{k+1}$ and $\tilde{\mathcal{J}}^{k+1}$ are stochastic versions of (6). All estimates are conditionally unbiased given the history and projectors.

Compatibility with momentum and adaptive optimizers. The **BROS** correction is orthogonal to the choice of optimizer: it only constructs the stochastic directions used by these two updates. Therefore, the same projected lower direction and auxiliary direction can be plugged into momentum SGD [52] or Adam-family optimizers [30, 18] by replacing the plain SGD updates for $\boldsymbol{Y}$ and $\boldsymbol{Z}$ with the corresponding optimizer steps. Appendix B.2 gives a more detailed discussion of this compatibility.

3 Convergence analysis for BROS

3.1 Assumptions

The first assumption is standard for hypergradient evaluation in bilevel analysis [7, 6, 32].

Assumption 1 (Smoothness, boundedness, and lower-level curvature)**.** *There exist positive constants $\mu_g, L_{f,0}, L_{f,1}, L_{g,1}, L_{g,2} > 0$ such that:*

1. *∇f and ∇g are Lipschitz continuous in $(x, \boldsymbol{Y})$ with constants $L_{f,1}$ and $L_{g,1}$, and $\nabla^2 g$ is $L_{g,2}$-Lipschitz continuous as a linear operator on the product space $\mathbb{R}^{d_x} \times \mathcal{Y}$;*
2. *$\|\nabla_{\boldsymbol{Y}} f(x, \boldsymbol{Y}^*(x))\| \leq L_{f,0}$ for all x;*
3. *for every fixed x, $g(x, \cdot)$ is μ_g-strongly convex on $\mathcal{Y}$.*

Let $L_{\max} := \max\{L_{f,0}, L_{f,1}, L_{g,1}, L_{g,2}\}$ and $\kappa := L_{\max}/\mu_g$.

Remark 1 (Scope of the theory)**.** *Our convergence analysis assumes lower-level strong convexity, as also adopted in bilevel methods with LLM experiments [45, 51]. Although general neural-network*

training need not satisfy this assumption, our nonconvex experiments empirically support the practical effectiveness of BROS beyond this theoretical regime.

The next assumption is the standard SBO oracle condition, adapted to the projected stochastic oracles used by **BROS**. A complete filtration-level statement is given in Appendix C.

Assumption 2 (Projected stochastic oracles)**.** *At iteration k, let $\mathcal{F}_k$ be the filtration before drawing $\boldsymbol{P}^k$ and let $\mathcal{F}_k^+$ additionally include $\boldsymbol{P}^k$. For the projected local objectives induced by $\boldsymbol{Y}^k + \boldsymbol{P}^k\boldsymbol{B}$, the stochastic oracles in Algorithm 1 satisfy:*

1. *the projected gradient oracles of f and g are conditionally unbiased given $\mathcal{F}_k^+$, and their estimation errors have uniformly bounded conditional second moments;*
2. *the projected HVP and JVP oracles are conditionally unbiased given $\mathcal{F}_k^+$, with conditional second moments of their respective estimation errors uniformly bounded by a constant times the squared norm of the queried direction;*
3. *multiple queries to the same sampled HVP/JVP operator within one iteration are allowed; the Rademacher probes are conditionally independent of the stochastic oracle samples given $\mathcal{F}_k^+$.*

The final assumption specifies the random subspaces used by **BROS**. It matches the standard isotropic projector assumption used in RSO [8], and its moment identities are used in the diagonal-block bias correction.

Assumption 3 (Random projectors and probes)**.** *Conditional on $\mathcal{F}_k$, the layerwise projectors $\{P_\ell^k\}_{\ell=1}^L$ are independent, with $P_\ell^k = \sqrt{m_\ell/r_\ell}\, O_{\ell,r_\ell}^k$, where O_{ℓ,r_ℓ}^k contains the first r_ℓ columns of a Haar-uniform matrix O_ℓ^k in the orthogonal group $O(m_\ell)$. We assume $r_\ell \geq 2$ for all ℓ. The Rademacher probes in Algorithm 1 are independent across layers and independent of the stochastic oracle samples conditional on $\mathcal{F}_k^+$.*

Projector sampling in implementation The Haar projector in Assumption 3 can be sampled without forming a full $m_\ell \times m_\ell$ orthogonal matrix. In implementation, for each layer we draw a Gaussian matrix $G_\ell^k \in \mathbb{R}^{m_\ell \times r_\ell}$, compute its thin QR factorization $G_\ell^k = Q_\ell^k R_\ell^k$ with the standard positive-diagonal convention for R_ℓ^k, and set $P_\ell^k = \sqrt{m_\ell/r_\ell}\, Q_\ell^k$. The resulting Q_ℓ^k is Haar-uniform on the Stiefel manifold, equivalently distributed as the first r_ℓ columns of a Haar-uniform matrix. The Stiefel manifold consists of matrices with orthonormal columns.

3.2 Convergence results for BROS

In this subsection, we state the main guarantee supporting the core claims of **BROS**. We first present the following lemma to state that **BROS** restores the same conditional expectations that a full-space single-loop method would use, while keeping the variance controlled. A detailed version with explicit constants is proved in Appendix C.

Lemma 1 (Moment preservation of **BROS** directions)**.** *Suppose Assumptions 1–3 hold. Define the full-space bilevel directions*

$$D_Y^k := \nabla_{\boldsymbol{Y}} g(x^k, \boldsymbol{Y}^k), \quad D_Z^k := \mathcal{H}(x^k, \boldsymbol{Y}^k)[\boldsymbol{Z}^k] - \nabla_{\boldsymbol{Y}} f(x^k, \boldsymbol{Y}^k), \quad D_x^k := \nabla_x f(x^k, \boldsymbol{Y}^k) - \mathcal{J}(x^k, \boldsymbol{Y}^k)[\boldsymbol{Z}^k].$$

For the analysis, denote the three stochastic directions generated by Algorithm 1 as $\widehat{\boldsymbol{V}}^{k+1} := \boldsymbol{P}^k \tilde{\boldsymbol{V}}^{k+1}$, $\widehat{\boldsymbol{S}}^{k+1} := \widehat{\mathcal{H}[\boldsymbol{Z}]}^{k+1} - \boldsymbol{P}^k \tilde{\boldsymbol{U}}_Y^{k+1}$, and $\widehat{W}^{k+1} := \tilde{U}_x^{k+1} - \tilde{\mathcal{J}}^{k+1}[\boldsymbol{P}^{k\top} \boldsymbol{Z}^k]$. Then

$$\mathbb{E}[\widehat{\boldsymbol{V}}^{k+1} \mid \mathcal{F}_k] = D_Y^k, \quad \mathbb{E}[\widehat{\boldsymbol{S}}^{k+1} \mid \mathcal{F}_k] = D_Z^k, \quad \mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] = D_x^k. \tag{11}$$

Moreover, there exists a finite constant $C_m > 0$, independent of K, such that

$$\mathbb{E}\Big[\|\widehat{\boldsymbol{V}}^{k+1} - D_Y^k\|^2 + \|\widehat{\boldsymbol{S}}^{k+1} - D_Z^k\|^2 + \|\widehat{W}^{k+1} - D_x^k\|^2 \,|\, \mathcal{F}_k\Big] \leq C_m \Big(1 + \|\boldsymbol{Y}^k - \boldsymbol{Y}^*(x^k)\|^2 + \|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2\Big). \tag{12}$$

Consequently, Lemma 1 allows us to view the projected recursions of **BROS** as full-space single-loop bilevel updates with unbiased directions and controlled projection-induced variance, which leads to the following convergence guarantee.

Theorem 1 (Main guarantee of **BROS**)**.** *Suppose Assumptions 1–3 hold, and assume also that $\Phi^* := \inf_x \Phi(x) > -\infty$. Consider Algorithm 1. Let $\omega_P := \max_{1 \leq \ell \leq L}(m_\ell/r_\ell)$ be the worst*

layerwise full-rank to subspace-rank ratio. Then there exist positive coupling constants c_1, c_2, c_3 and finite constants $C_0, C_1, \bar{\alpha} > 0$, such that the following holds: If $\alpha_k \equiv \alpha$, $\beta_k \equiv c_1\alpha$, $\gamma_k \equiv c_2\alpha$, $\theta_k \equiv c_3\alpha$, with $0 < \alpha \leq (\bar{\alpha}/(1+\omega_P^3))$, we have (see detailed proof in Appendix C):

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla\Phi(x^k)\|^2 \leq 2\left(\frac{C_0(1+\omega_P^2)}{K\alpha} + C_1(1+\omega_P^4)\alpha\right). \tag{13}$$

Choosing $\alpha = \Theta(K^{-1/2})$ subject to the above upper bound yields

$$\mathbb{E}\|\nabla\Phi(x^R)\|^2 = \mathcal{O}\Big((1+\omega_P^4)K^{-1/2}\Big),$$

where $R \sim \mathrm{Unif}\{0,\ldots,K-1\}$ is independent of the algorithmic randomness. Hence, for fixed projection ratio ω_P, **BROS** *attains the standard $\mathcal{O}(\varepsilon^{-2})$ sample complexity for finding an ε-stationary point. With the projection factor displayed, it suffices to take $K = \mathcal{O}((1+\omega_P^4)^2\varepsilon^{-2})$.*

Full-space-order rate and projection-ratio dependence. Theorem 1 separates the standard single-loop SBO order from the price of projection: the dependence on K matches the $\mathcal{O}(K^{-1/2})$ stationarity rate of MA-SOBA with full-space hypergradient evaluation [7], while the constants depend on $\omega_P = \max_\ell m_\ell/r_\ell$. Larger subspace ranks decrease ω_P, improving both the transient term $C_0(1+\omega_P^2)/(K\alpha)$ and the stochastic-error term $C_1(1+\omega_P^4)\alpha$, and allowing a larger stepsize through $\alpha \leq \bar{\alpha}/(1+\omega_P^3)$. Conversely, smaller ranks save memory and projected-oracle cost but inflate the variance of lifted stochastic directions, so the same stationarity level may require more iterations or a smaller stepsize. In the full-rank case $r_\ell = m_\ell$ for all layers, $\omega_P = 1$, so the result recovers the usual full-space single-loop stochastic bilevel rate up to absolute constants.

Role of the bias correction. The above MA-SOBA-order guarantee relies on the **BROS** correction because it centers the auxiliary linear-system recursion at the full-space Hessian action, which is required by Lemma 1 to target the true hypergradient $\nabla\Phi(x)$. Appendix C.5 gives a deterministic quadratic counterexample showing that, without this correction, the naive projected auxiliary HVP can converge to a biased auxiliary fixed point and drive the upper update to a nonstationary point, even with no stochastic objective noise and exact lower-level solutions.

4 Memory analysis

The memory savings of **BROS** stem directly from the randomized-subspace computations in Algorithm 1. Under the peak-memory proxy considered here, the projected lower- and auxiliary-level computations reduce both the trainable-side activation footprint and the online lower/auxiliary directions from full-size tensors to subspace-size tensors. The auxiliary HVP/JVP queries are also evaluated in projected coordinates, reducing the backend-dependent memory pressure of second-order autodifferentiation; accordingly, we report this component by query scale rather than folding it into a backend-specific closed-form constant. Notably, RSO reports that activations can dominate trainable-side memory in large-batch regimes [8], making **BROS** 's activation savings important when the lower-level task trains neural networks.

We analyze one trainable Transformer decoder block with a SwiGLU feed-forward network [49] and standard multi-head attention [55], i.e., without grouped-query attention [1] (non-GQA). The component-wise derivation is provided in Appendix D.2. Let n denote the hidden size, s the sequence length, b the micro-batch size, and h the number of attention heads. We count scalar memory slots, omit upper-level variables and datatype constants, and use a peak-memory proxy that includes persistent lower/auxiliary states, saved activations, and online gradient/direction tensors. Under this accounting, the counted peak-memory proxy of **BROS** is

$$M_{\mathrm{peak}}^{\mathbf{BROS}} = 24n^2\ (\text{state}) + \frac{28}{3}bsn\ (\text{hidden}) + 2bhs^2\ (\text{attn}) + 4bsr\ (\text{proj-act}) + \frac{62}{3}rn\ (\text{grad/dir}).$$

Table 2 gives the memory comparison between **BROS** and MA-SOBA [7], FdeHBO [63], and Penalty [42]. The complete derivation is provided in Appendix D.3.

Table 2: Non-GQA trainable-side peak-memory proxy for one decoder block, counted in scalar slots and excluding upper variables. The HVP/JVP column reports the derivative-query scale because its exact peak depends on the autodiff backend.

Method	Lower/aux. states	Activations	Grad./directions	HVP/JVP scale	Peak proxy
MA-SOBA [7]	$24n^2$	$15bsn + 2bhs^2$	$24n^2$	full-space HVP/JVP	$48n^2 + 15bsn + 2bhs^2$
FdeHBO [63]	$48n^2$	$15bsn + 2bhs^2$	$24n^2$	full-space finite-difference directions	$72n^2 + 15bsn + 2bhs^2$
Penalty [42]	$12n^2$	$15bsn + 2bhs^2$	$12n^2$	no auxiliary HVP/JVP	$24n^2 + 15bsn + 2bhs^2$
BROS	$24n^2$	$\frac{28}{3}bsn + 2bhs^2 + 4bsr$	$\frac{62}{3}rn$	projected HVP/JVP	$24n^2 + \frac{28}{3}bsn + 2bhs^2 + 4bsr + \frac{62}{3}rn$

Let $\rho := r/n$ and $\tau := bs/n$. Compared with MA-SOBA in Table 2, **BROS** reduces both the activation term and the online gradient/direction term for any nontrivial subspace ratio $\rho < 1$; for example, at $(\tau, \rho) = (1, 0.25)$, the displayed peak-memory proxy is reduced by about 37.3%. Penalty is the strongest memory-oriented surrogate baseline in the table, since it avoids the auxiliary linear system and its HVP/JVP queries. Even against Penalty, **BROS** can be more memory efficient in activation-dominated regimes; for example, at $(\tau, \rho) = (2, 0.25)$, it saves about 7.7% in the displayed proxy. Unlike Penalty, however, **BROS** preserves the exact single-loop hypergradient structure and the MA-SOBA-order convergence guarantee. These numbers do not include the additional peak-memory reduction from replacing full-space HVP/JVP autodiff queries by projected ones.

5 Experiments

We evaluate **BROS** on four bilevel learning tasks: MNIST hyper-data cleaning, data-mixture learning for language-model training, CIFAR-10 hyper-representation learning, and CIFAR-100 ViT sample reweighting. Across all experiments, **BROS** uses randomized subspace updates with a projection ratio $r/n = 0.25$ unless otherwise specified. The optimizer used for each BROS variable and training stage is specified in Table 5 of Appendix E.

Hyper-data cleaning. We follow the convex MNIST data-cleaning setting in [48]. The training labels are corrupted by 50% random label noise, while clean validation and test sets are used for the upper-level objective and final evaluation. We compare Penalty [42], FdeHBO [63], MA-SOBA [7], ZOFO [51], and **BROS**. Figure 1 shows that **BROS** reaches 84.30% final test accuracy, closely matching MA-SOBA at 84.46% and outperforming Penalty, FdeHBO, and ZOFO. The ablation further shows that projection ratios from $r/n = 0.10$ to $r/n = 0.50$ retain strong performance, while an extremely small ratio $r/n = 0.01$ degrades accuracy slightly.

Data-mixture learning. We also evaluate data-mixture learning on 17 copyright-free Pile domains. A 280M GPT-style proxy model first learns the data-mixture weights, and a 280M main model is then trained from scratch with the learned mixture. Both proxy and main models use 16 layers, hidden size 1024, and 16 heads with the EleutherAI/gpt-neox-20b [4] tokenizer. Original Pile Mixture uses the original domain proportions, without proxy-learned weights. Figure 2 shows the main-model average evaluation-loss curves. **BROS** achieves average loss 2.8109, essentially matching MA-SOBA (2.8098) and ZOFO (2.8114), while avoiding the unstable domain concentration observed for DoReMi [58]. DoReMi requires predefined domains and reference losses, so we compare it only in this task.

The proxy-memory profiling results are included in the legend of Figure 2. **BROS** reduces peak memory by about 45% relative to MA-SOBA and about 27% relative to Penalty, while retaining a substantially higher step rate than ZOFO.

Hyper-representation learning. We further test **BROS** on CIFAR-10 hyper-representation learning, where the upper-level objective evaluates the learned representation and the lower-level problem trains predictors under that representation. We compare with the same bilevel baselines as above and report the full accuracy curves and projection-ratio ablation in Figure 3 of Appendix E.

ViT sample reweighting. We also evaluate a larger CIFAR-100 sample-reweighting task with 20% label noise, where the lower model is a small ViT trained from scratch and the upper level selects sample weights using a clean validation split. Detailed settings and the validation, test, and peak-memory comparison are given in Table 12 of Appendix E. Appendix E.5–E.7 reports additional bi-probe ablations, timing profiles, and wall-clock comparisons.

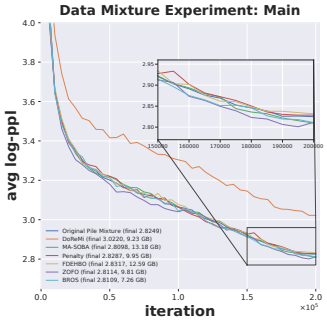


(a) Main comparison.



(b) Projection-ratio ablation.

Figure 1: **Hyper-data cleaning on MNIST with 50% label noise.** **(a)** Main comparison among Penalty, FdeHBO, MA-SOBA, ZOFO, and **BROS** with projection ratio $r/n = 0.25$. **(b)** Projection-ratio ablation for **BROS**.



(a) Main-model evaluation.

Figure 2: **Data-mixture learning.** Main-model evaluation after training with proxy-learned data mixtures.

6 Conclusion and future work

We propose **BROS**, a memory-efficient randomized-subspace method for stochastic bilevel optimization with multilayer matrix lower-level variables. By decomposing the lower and auxiliary recursions into randomized subspace updates and correcting the projection-induced auxiliary-HVP bias with Rademacher bi-probes, **BROS** preserves an unbiased hypergradient estimator and matches the $\mathcal{O}(\varepsilon^{-2})$ convergence order of MA-SOBA while reducing trainable-side memory. Future work includes further reducing the remaining activation memory in large architectures and extending the convergence analysis beyond the unpreconditioned recursion analyzed here.

Acknowledgments and Disclosure of Funding

This work is funded by the National Natural Science Foundation of China (No. 12288101, W2441021), and the National Key Research and Development Program of China (No. 2024YFA1012902). This research is also supported by the AI for Science Institute, Beijing, China. Part of the computational resources were supported by the Center for Intelligent Computing and Song-Shan Lake HPC Center (SSL-HPC) in Great Bay University, Dongguan, China.

The authors declare no competing interests.

References

[1] J. Ainslie, J. Lee-Thorp, M. De Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, 2023.

[2] M. Arbel and J. Mairal. Amortized implicit differentiation for stochastic bilevel optimization. *arXiv preprint arXiv:2111.14580*, 2021.

[3] L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

[4] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, M. Pieler, U. S. Prashanth, S. Purohit, L. Reynolds, J. Tow, B. Wang, and S. Weinbach. GPT-NeoX-20B: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

[5] T. Chen, Y. Sun, and W. Yin. Tighter analysis of alternating stochastic gradient method for stochastic nested problems. *arXiv preprint arXiv:2106.13781*, 2021.

[6] X. Chen, M. Huang, S. Ma, and K. Balasubramanian. Decentralized stochastic bilevel optimization with improved per-iteration complexity. In *International Conference on Machine Learning*, pages 4641–4671. PMLR, 2023.

[7] X. Chen, T. Xiao, and K. Balasubramanian. Optimal algorithms for stochastic bilevel optimization under relaxed smoothness conditions. *Journal of Machine Learning Research*, 25(151):1–51, 2024.

[8] Y. Chen, Y. Zhang, Y. Liu, K. Yuan, and Z. Wen. A memory efficient randomized subspace optimization method for training large language models. *arXiv preprint arXiv:2502.07222*, 2025.

[9] T. Chu, D. Xu, W. Yao, and J. Zhang. SPABA: A single-loop and probabilistic stochastic bilevel algorithm achieving optimal sample complexity. *arXiv preprint arXiv:2405.18777*, 2024.

[10] M. Dagréou, P. Ablin, S. Vaiter, and T. Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. *Advances in Neural Information Processing Systems*, 35:26698–26710, 2022.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[12] S. Fan, M. Pagliardini, and M. Jaggi. Doge: Domain reweighting with generalization estimation. *arXiv preprint arXiv:2310.15393*, 2023.

[13] L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, and M. Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR, 2018.

[14] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[15] S. Ghadimi, A. Ruszczynski, and M. Wang. A single timescale stochastic approximation method for nested stochastic optimization. *SIAM Journal on Optimization*, 30(1):960–979, 2020.

[16] S. Ghadimi and M. Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

[17] R. Gower, D. Kovalev, F. Lieder, and P. Richtárik. RSN: randomized subspace newton. *Advances in Neural Information Processing Systems*, 32, 2019.

[18] Z. Guo, Y. Xu, W. Yin, R. Jin, and T. Yang. A novel convergence analysis for algorithms of the adam family. *arXiv preprint arXiv:2112.03459*, 2021.

[19] A. Han, J. Li, W. Huang, M. Hong, A. Takeda, P. Jawanpuria, and B. Mishra. SLTrain: a sparse plus low rank approach for parameter and memory efficient pretraining. *Advances in Neural Information Processing Systems*, 37:118267–118295, 2024.

[20] J. Hao, X. Gong, and M. Liu. Bilevel optimization under unbounded smoothness: A new algorithm and convergence analysis. *arXiv preprint arXiv:2401.09587*, 2024.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[22] Y. He, J. Hu, X. Huang, S. Lu, B. Wang, and K. Yuan. Distributed bilevel optimization with communication compression. *arXiv preprint arXiv:2405.18858*, 2024.

[23] Y. He, P. Li, Y. Hu, C. Chen, and K. Yuan. Subspace optimization for large language models with convergence guarantees. *arXiv preprint arXiv:2410.11289*, 2024.

[24] M. Hong, H.-T. Wai, Z. Wang, and Z. Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180, 2023.

[25] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

[26] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[27] F. Huang. Optimal hessian/jacobian-free nonconvex-pl bilevel optimization. *arXiv preprint arXiv:2407.17823*, 2024.

[28] K. Ji, J. Yang, and Y. Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International Conference on Machine Learning*, pages 4882–4892. PMLR, 2021.

[29] P. Khanduri, S. Zeng, M. Hong, H.-T. Wai, Z. Wang, and Z. Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in Neural Information Processing Systems*, 34:30271–30283, 2021.

[30] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] B. Kong, J. Liang, Y. Liu, R. Deng, and K. Yuan. Cr-net: Scaling parameter-efficient training with cross-layer low-rank structure. *arXiv preprint arXiv:2509.18993*, 2025.

[32] B. Kong, S. Zhu, S. Lu, X. Huang, and K. Yuan. Decentralized bilevel optimization: A perspective from transient iteration complexity. *Journal of Machine Learning Research*, 26(240):1–64, 2025.

[33] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[34] J. Kwon, D. Kwon, S. Wright, and R. Nowak. On penalty methods for nonconvex bilevel optimization and first-order stochastic approximation. *arXiv preprint arXiv:2309.01753*, 2023.

[35] J. Kwon, D. Kwon, S. Wright, and R. D. Nowak. A fully first-order method for stochastic bilevel optimization. In *International Conference on Machine Learning*, pages 18083–18113. PMLR, 2023.

[36] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online].*, 2, 2010.

[37] J. Li, B. Gu, and H. Huang. A fully single loop algorithm for bilevel optimization without hessian inverse. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7426–7434, 2022.

[38] J. Li, F. Huang, and H. Huang. Communication-efficient federated bilevel optimization with local and global lower level problems. *arXiv preprint arXiv:2302.06701*, 2023.

[39] V. Lialin, N. Shivagunde, S. Muckatira, and A. Rumshisky. Relora: High-rank training through low-rank updates. *arXiv preprint arXiv:2307.05695*, 2023.

[40] S.-Y. Liu, C.-Y. Wang, H. Yin, P. Molchanov, Y.-C. F. Wang, K.-T. Cheng, and M.-H. Chen. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024.

[41] Z. Liu, R. Zhang, Z. Wang, M. Yan, Z. Yang, P. D. Hovland, B. Nicolae, F. Cappello, S. Tang, and Z. Zhang. CoLA: Compute-efficient pre-training of LLMs via low-rank activation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4627–4645, 2025.

[42] A. Mehra and J. Hamm. Penalty method for inversion-free deep bilevel optimization. In *Asian conference on machine learning*, pages 347–362. PMLR, 2021.

[43] R. Miles, P. Reddy, I. Elezi, and J. Deng. Velora: Memory efficient training using rank-1 sub-token projections. *Advances in Neural Information Processing Systems*, 37:42292–42310, 2024.

[44] Z. Mo, L.-K. Huang, and S. J. Pan. Parameter and memory efficient pretraining via low-rank riemannian optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

[45] R. Pan, D. Zhang, H. Zhang, X. Pan, M. Xu, J. Zhang, R. Pi, X. Wang, and T. Zhang. ScaleBiO: Scalable bilevel optimization for LLM data reweighting. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31959–31982, 2025.

[46] A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine. Meta-learning with implicit gradients. *Advances in Neural Information Processing Systems*, 32, 2019.

[47] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pages 4334–4343. PMLR, 2018.

[48] A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1723–1732. PMLR, 2019.

[49] N. Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[50] H. Shen, P.-Y. Chen, P. Das, and T. Chen. SEAL: Safety-enhanced aligned LLM fine-tuning via bilevel data selection. *arXiv preprint arXiv:2410.07471*, 2024.

[51] R. Shirkavand, P. Yu, Q. He, and H. Huang. Bilevel ZOFO: Efficient LLM fine-tuning and meta-training. *arXiv preprint arXiv:2502.03604*, 2025.

[52] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013.

[53] D. A. Tarzanagh, M. Li, C. Thrampoulidis, and S. Oymak. Fednest: Federated bilevel, minimax, and compositional optimization. In *International Conference on Machine Learning*, pages 21146–21179. PMLR, 2022.

[54] Y. Tu, B. Zhang, Y. Li, L. Liu, J. Li, Y. Wang, C. Wang, and C. R. Zhao. Learning from noisy labels with decoupled meta label purifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19934–19943, 2023.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[56] R.-N. Wang, Y.-Q. Qiao, Z.-L. Xie, and Y. Kun. DRAW: Domain weight randomization with bayesian updating for LLM pre-training. *Transactions on Machine Learning Research*, 2026.

[57] J. Xiao, S. Sang, T. Zhi, J. Liu, Q. Yan, L. Luo, and B. Yuan. Coap: Memory-efficient training with correlation-aware gradient projection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 30116–30126, 2025.

[58] S. M. Xie, H. Pham, X. Dong, N. Du, H. Liu, Y. Lu, P. S. Liang, Q. V. Le, T. Ma, and A. W. Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023.

[59] J. Yang, K. Ji, and Y. Liang. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34:13670–13682, 2021.

[60] S. Yang, X. Zhang, and M. Wang. Decentralized gossip-based stochastic bilevel optimization over communication networks. *Advances in Neural Information Processing Systems*, 35:238–252, 2022.

[61] Y. Yang, H. Ban, M. Huang, S. Ma, and K. Ji. Tuning-free bilevel optimization: New algorithms and convergence analysis. *arXiv preprint arXiv:2410.05140*, 2024.

[62] Y. Yang, B. Gao, and Y.-x. Yuan. Lancbio: dynamic lanczos-aided bilevel optimization via krylov subspace. *arXiv preprint arXiv:2404.03331*, 2024.

[63] Y. Yang, P. Xiao, and K. Ji. Achieving $\mathcal{O}(\epsilon^{-1.5})$ complexity in hessian/jacobian-free stochastic bilevel optimization. *Advances in Neural Information Processing Systems*, 36:39491–39503, 2023.

[64] Y. Yang, P. Xiao, and K. Ji. Simfbo: Towards simple, flexible and communication-efficient federated bilevel learning. *Advances in Neural Information Processing Systems*, 36:33027–33040, 2023.

[65] Y. Yang, P. Xiao, S. Ma, and K. Ji. First-order federated bilevel learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22029–22037, 2025.

[66] H. Zhang, J. Yin, G. Wang, Z. Liu, L. F. Yang, T. Zhang, A. Shrivastava, and V. Braverman. Breaking the frozen subspace: Importance sampling for low-rank optimization in LLM pretraining. *arXiv preprint arXiv:2502.05790*, 2025.

[67] J. Zhao, Z. Zhang, B. Chen, Z. Wang, A. Anandkumar, and Y. Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

[68] G. Zheng, A. H. Awadallah, and S. Dumais. Meta label correction for noisy label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11053–11061, 2021.

[69] H. Zhu, Z. Zhang, W. Cong, X. Liu, S. Park, V. Chandra, B. Long, D. Z. Pan, Z. Wang, and J. Lee. Apollo: SGD-like memory, adamw-level performance. *Proceedings of Machine Learning and Systems*, 7, 2025.

[70] S. Zhu, B. Kong, S. Lu, X. Huang, and K. Yuan. SPARKLE: a unified single-loop primal-dual framework for decentralized bilevel optimization. *Advances in Neural Information Processing Systems*, 37:62912–62987, 2024.

Appendix for "BROS: Bias-Corrected Randomized Subspaces for Memory-Efficient Single-Loop Bilevel Optimization"

Contents

1 Introduction **1**

2 Stochastic bilevel optimization via randomized subspace (BROS) **4**

2.1 Hypergradient computation and single-loop bilevel updates 4

2.2 Randomized subspace sketching for bilevel recursions 5

2.3 Rademacher-probe recovery of auxiliary HVPs 5

3 Convergence analysis for BROS **6**

3.1 Assumptions 6

3.2 Convergence results for **BROS** 7

4 Memory analysis **8**

5 Experiments **9**

6 Conclusion and future work **10**

A Related work **16**

B Additional algorithm discussion **16**

B.1 Multilayer derivation of the **BROS** recursion 16

B.2 Compatibility with momentum and adaptive optimizers 17

C Theoretical analysis **18**

C.1 Assumptions 18

C.2 Proof roadmap 19

C.3 Supporting lemmas 20

C.4 Main convergence theorem 38

C.5 A quadratic counterexample for naive projected auxiliary HVPs 40

D Memory analysis for Non-GQA Transformer blocks **41**

D.1 Transformer layer and memory components 41

D.2 BROS memory in the non-GQA case 42

D.3 Memory of other bilevel baselines 42

E Additional experiments **43**

E.1 MNIST hyper-data cleaning 43
E.2 Data-mixture learning 44
E.3 Hyper-representation learning 44
E.4 CIFAR-100 ViT sample reweighting 45
E.5 Bi-probe ablation 46
E.6 Component and per-step timing 46
E.7 Performance under wall-clock budgets 47
E.8 Licenses of used datasets and models 48

F Broader impacts **48**

A Related work

Stochastic bilevel optimization. Bilevel optimization is widely used in hyperparameter optimization, meta-learning, data reweighting, and representation learning [13, 46, 3, 47, 27]. The main technical difficulty is to estimate the hypergradient, since the upper-level gradient depends on how the lower-level solution changes with the upper-level variable. Early stochastic methods use double-loop updates, truncated backpropagation, approximate implicit differentiation, or two-timescale stochastic approximation [16, 24, 28, 2, 5]. More recent single-loop methods update the lower variable, an auxiliary variable, and the upper variable in one loop. Representative examples include the single-timescale method [15], SUSTAIN [29], SOBA/SABA [10], MA-SOBA [7], and SPABA [9]. Beyond the single-machine setting, decentralized and federated bilevel methods extend the same nested optimization paradigm to networked or client-server systems, where hypergradient estimation interacts with communication topology, client participation, data heterogeneity, and consensus errors [60, 6, 53, 38, 64, 32, 70, 65, 22]. These methods are the closest baseline for our work, but they still form both the lower update and the auxiliary update in the full parameter space.

Bilevel methods with alternative hypergradient estimators or formulations. Another line of work studies bilevel optimization by changing how the hypergradient is computed, by adopting a different oracle model, or by reformulating the original bilevel problem. Penalty methods [42, 34] replace the original bilevel problem with a penalty reformulation. Fully first-order methods [37, 35] avoid Hessian- and Jacobian-vector products. FdeHBO [63] uses finite differences to avoid explicit Hessian- and Jacobian-vector products. Bilevel-ZOFO [51] combines zeroth-order outer updates with parameter-efficient inner updates. These methods reduce the burden of full hypergradient computation, but they no longer follow the same standard single-loop hypergradient update as methods such as MA-SOBA [7]: they either optimize a surrogate problem, use a different oracle, or rely on weaker hypergradient information.

Memory-efficient subspace and low-rank training. Subspace and projection methods are a common way to reduce memory in single-level neural-network optimization. A broad class of methods forms updates in lower-dimensional spaces, either by projecting gradients or optimizer states [67, 66, 23, 57, 69], or by restricting learning to sampled low-dimensional subspaces, as in randomized subspace optimization (RSO) [8]. Related parameter-efficient tuning methods also reduce the number of trainable degrees of freedom through low-dimensional modules, adapters, prompts, or low-rank updates [26, 39, 40, 19, 43, 25, 41, 31, 44]. Subspace ideas also appear in bilevel optimization, but with a different goal: LancBiO [62] uses Krylov/Lanczos subspaces to approximate Hessian-inverse-vector products by solving small projected linear systems. This reduces the computational cost of the auxiliary linear solve, but it is not designed to reduce trainable-side memory; maintaining Krylov bases and projected matrices can introduce additional storage. These works motivate our study, but they do not consider randomized subspace projection within single-loop bilevel hypergradient methods. In particular, they do not address how such projections affect the auxiliary Hessian-vector computation and the resulting hypergradient estimate.

Neural bilevel applications. Our matrix-block setting is motivated by neural bilevel problems in which the lower-level variable is itself a multilayer model. Typical examples include data hyper-cleaning [48], sample reweighting and label correction [47, 68, 54, 20, 61], hyper-representation learning [13, 27, 65], and meta-learning [46, 3]. Recent data-selection and data-mixture methods for larger models were studied in [12, 58, 50, 45, 56]. These applications motivate our problem setting, but our contribution is algorithmic: we study how to reduce the memory used by stochastic bilevel optimization while keeping the hypergradient estimator unbiased.

B Additional algorithm discussion

B.1 Multilayer derivation of the BROS recursion

Section 2 introduces the main idea through the single-layer case. We now spell out the corresponding multilayer notation used in Algorithm 1. Recall that $\boldsymbol{Y} = (Y_1, \dots, Y_L)$ with $Y_\ell \in \mathbb{R}^{m_\ell \times n_\ell}$. At iteration k, **BROS** samples independent scaled Haar projectors $P_\ell^k \in \mathbb{R}^{m_\ell \times r_\ell}$ and writes $\boldsymbol{P}^k = (P_1^k, \dots, P_L^k)$. The reduced search variable is $\boldsymbol{B} = (B_1, \dots, B_L) \in \mathcal{B} := \prod_{\ell=1}^{L} \mathbb{R}^{r_\ell \times n_\ell}$, and blockwise multiplication is defined by $(\boldsymbol{P}^k \boldsymbol{B})_\ell = P_\ell^k B_\ell$ and $(\boldsymbol{P}^{k\top} \boldsymbol{U})_\ell = (P_\ell^k)^\top U_\ell$.

The multilayer RSO lower-level step is obtained from the local projected objective

$$\tilde{g}_k(x, \boldsymbol{B}) := g(x, \boldsymbol{Y}^k + \boldsymbol{P}^k \boldsymbol{B}).$$

Thus the lower direction is formed in the reduced product space $\mathcal{B}$ and then lifted back to the matrix-block space as $\boldsymbol{P}^k \tilde{\boldsymbol{V}}^{k+1}$, where $\tilde{\boldsymbol{V}}^{k+1}$ estimates $\nabla_{\boldsymbol{B}} \tilde{g}_k(x^k, \boldsymbol{0})$. Similarly, $\tilde{\boldsymbol{U}}_Y^{k+1}$ estimates $\nabla_{\boldsymbol{B}} \tilde{f}_k(x^k, \boldsymbol{0})$, $\tilde{U}_x^{k+1}$ estimates $\nabla_x \tilde{f}_k(x^k, \boldsymbol{0})$, and the projected linear operators $\tilde{\mathcal{H}}^{k+1} : \mathcal{B} \to \mathcal{B}$ and $\tilde{\mathcal{J}}^{k+1} : \mathcal{B} \to \mathbb{R}^{d_x}$ estimate the projected HVP and JVP.

The auxiliary HVP is assembled blockwise. Let $E_t(W) \in \mathcal{B}$ denote the reduced block variable that inserts $W \in \mathbb{R}^{r_t \times n_t}$ into block t and zeros elsewhere. For a target layer ℓ, the full Hessian action decomposes as $(\mathcal{H}^k[\boldsymbol{Z}^k])_\ell = \sum_{t=1}^L \mathcal{H}_{\ell t}^k[Z_t^k]$. The off-diagonal contribution from block $t \neq \ell$ is lifted directly from the projected HVP:

$$A_\ell^{(t),k+1} := P_\ell^k \big(\tilde{\mathcal{H}}^{k+1}[E_t((P_t^k)^\top Z_t^k)]\big)_\ell.$$

For the diagonal contribution, the one-sided row projection creates the same self-block distortion as in the single-layer case. Therefore **BROS** applies the single-layer correction in (10) layerwise:

$$\widehat{\mathcal{H}_{\ell\ell}[Z_\ell]}^{k+1} := \mathrm{CorrHVP}_\ell\Big(P_\ell^k, Z_\ell^k, \tilde{\mathcal{H}}^{k+1}; u_\ell^{k+1}, \xi_\ell^{k+1}\Big).$$

The full multilayer estimator used in Algorithm 1 is then

$$(\widehat{\mathcal{H}[\boldsymbol{Z}]}^{k+1})_\ell := \widehat{\mathcal{H}_{\ell\ell}[Z_\ell]}^{k+1} + \sum_{t \neq \ell} A_\ell^{(t),k+1}, \quad \ell = 1, \dots, L.$$

This estimator is plugged into the auxiliary residual $\widehat{\mathcal{H}[\boldsymbol{Z}]}^{k+1} - \boldsymbol{P}^k \tilde{\boldsymbol{U}}_Y^{k+1}$, while the projected mixed-Jacobian term $\tilde{\mathcal{J}}^{k+1}[\boldsymbol{P}^{k\top} \boldsymbol{Z}^k]$ enters the moving-average hypergradient estimator. The resulting coupled update is exactly the recursion stated in Algorithm 1. The moment-preservation proof for this blockwise construction is given in Lemma 5.

B.2 Compatibility with momentum and adaptive optimizers

BROS is an estimator-level construction rather than a complete prescription for the optimizer used to integrate the lower and auxiliary directions. At each iteration it produces two directions in the original matrix-block space:

$$G_Y^{k+1} := \boldsymbol{P}^k \tilde{\boldsymbol{V}}^{k+1}, \quad G_Z^{k+1} := \widehat{\mathcal{H}[\boldsymbol{Z}]}^{k+1} - \boldsymbol{P}^k \tilde{\boldsymbol{U}}_Y^{k+1}.$$

Algorithm 1 uses the plain stochastic steps $\boldsymbol{Y}^{k+1} = \boldsymbol{Y}^k - \beta_k G_Y^{k+1}$ and $\boldsymbol{Z}^{k+1} = \boldsymbol{Z}^k - \gamma_k G_Z^{k+1}$ because this is the recursion analyzed in Section 3. In implementation, these two directions can instead be passed to a momentum or adaptive optimizer.

For a momentum SGD-like implementation, maintain velocity variables $\boldsymbol{M}_Y^k$ and $\boldsymbol{M}_Z^k$ with the same block structure as $\boldsymbol{Y}$ and $\boldsymbol{Z}$. Given momentum parameters $\rho_Y, \rho_Z \in [0, 1)$, the lower and auxiliary updates can be written as

$$\boldsymbol{M}_Y^{k+1} := \rho_Y \boldsymbol{M}_Y^k + (1 - \rho_Y) G_Y^{k+1}, \quad \boldsymbol{Y}^{k+1} := \boldsymbol{Y}^k - \beta_k \boldsymbol{M}_Y^{k+1},$$
$$\boldsymbol{M}_Z^{k+1} := \rho_Z \boldsymbol{M}_Z^k + (1 - \rho_Z) G_Z^{k+1}, \quad \boldsymbol{Z}^{k+1} := \boldsymbol{Z}^k - \gamma_k \boldsymbol{M}_Z^{k+1}.$$

Thus the only difference from Algorithm 1 is that the projected lower direction and the corrected auxiliary direction are accumulated through momentum before updating the persistent variables.

For an Adam-like implementation, use G_Y^{k+1} and G_Z^{k+1} as gradient-like inputs to first- and second-moment recursions. For $\boldsymbol{Y}$, with moment parameters $\rho_1, \rho_2 \in [0, 1)$, numerical constant $\epsilon_{\mathrm{A}} > 0$, and decoupled weight decay $\lambda_Y \geq 0$, one may use

$$\boldsymbol{m}_Y^{k+1} := \rho_1 \boldsymbol{m}_Y^k + (1 - \rho_1) G_Y^{k+1}, \quad \boldsymbol{q}_Y^{k+1} := \rho_2 \boldsymbol{q}_Y^k + (1 - \rho_2)(G_Y^{k+1} \odot G_Y^{k+1}),$$
$$\widehat{\boldsymbol{m}}_Y^{k+1} := \frac{\boldsymbol{m}_Y^{k+1}}{1 - \rho_1^{k+1}}, \quad \widehat{\boldsymbol{q}}_Y^{k+1} := \frac{\boldsymbol{q}_Y^{k+1}}{1 - \rho_2^{k+1}}, \quad \boldsymbol{Y}^{k+1} := (1 - \beta_k \lambda_Y) \boldsymbol{Y}^k - \beta_k \frac{\widehat{\boldsymbol{m}}_Y^{k+1}}{\sqrt{\widehat{\boldsymbol{q}}_Y^{k+1}} + \epsilon_{\mathrm{A}}}.$$

The auxiliary variable is updated analogously:

$$\boldsymbol{m}_Z^{k+1} := \rho_1 \boldsymbol{m}_Z^k + (1-\rho_1) G_Z^{k+1}, \quad \boldsymbol{q}_Z^{k+1} := \rho_2 \boldsymbol{q}_Z^k + (1-\rho_2)(G_Z^{k+1} \odot G_Z^{k+1}),$$

$$\widehat{\boldsymbol{m}}_Z^{k+1} := \frac{\boldsymbol{m}_Z^{k+1}}{1-\rho_1^{k+1}}, \quad \widehat{\boldsymbol{q}}_Z^{k+1} := \frac{\boldsymbol{q}_Z^{k+1}}{1-\rho_2^{k+1}}, \quad \boldsymbol{Z}^{k+1} := (1-\gamma_k\lambda_Z)\boldsymbol{Z}^k - \gamma_k \frac{\widehat{\boldsymbol{m}}_Z^{k+1}}{\sqrt{\widehat{\boldsymbol{q}}_Z^{k+1}} + \epsilon_{\mathrm{A}}}.$$

All products, square roots, and divisions in these Adam-like formulas are understood elementwise within each matrix block. These choices do not change the Rademacher bi-probe formula or the blockwise corrected HVP estimator; they only change how the resulting directions are accumulated across iterations.

There are two implementation conventions. The most direct convention maintains optimizer states in the original matrix-block space and feeds the lifted directions G_Y^{k+1} and G_Z^{k+1} to the optimizer. This is simple and compatible with standard optimizer implementations. A more memory-oriented convention applies momentum or adaptive updates to the reduced variables inside the current projected subproblem. In that case, the optimizer state is tied to the sampled projector and must be reset, refreshed, or transported when the projector changes. This projected-state convention is natural for subspace training, but its exact behavior depends on the projector refresh schedule and the optimizer-state transport rule.

The convergence theorem in the main text analyzes the unpreconditioned recursion in Algorithm 1, including its EMA hypergradient tracker. The key unbiasedness property of the **BROS** directions remains an estimator property, but a convergence proof for momentum or adaptive variants would need additional assumptions controlling the optimizer states or preconditioners. We therefore treat these variants as implementation-compatible extensions and leave their nonasymptotic analysis to future work.

C Theoretical analysis

C.1 Assumptions

Assumption 4. *There exist constants $\mu_g, L_{f,0}, L_{f,1}, L_{g,1}, L_{g,2}$ such that:*

1. *∇f and ∇g are $L_{f,1}$- and $L_{g,1}$-Lipschitz continuous in $(x, \boldsymbol{Y})$, using Euclidean norm for x and the product Frobenius norm for $\boldsymbol{Y}$;*
2. *$\nabla^2 g$ is $L_{g,2}$-Lipschitz continuous as a block Hessian operator on $\mathbb{R}^{d_x} \times \mathcal{Y}$;*
3. *$\|\nabla_{\boldsymbol{Y}} f(x, \boldsymbol{Y}^*(x))\| \le L_{f,0}$ for all $x \in \mathbb{R}^{d_x}$;*
4. *for every fixed x, $\boldsymbol{Y} \mapsto g(x, \boldsymbol{Y})$ is μ_g-strongly convex in the product Frobenius norm.*

Moreover, define $L_{\max} := \max\{L_{f,0}, L_{f,1}, L_{g,1}, L_{g,2}\}$ and $\kappa := L_{\max}/\mu_g$.

Assumption 5 (Stochastic projected oracles and filtration)**.** *For any $k \ge 0$, let*

$$\mathcal{F}_k = \sigma\{x^0, \dots, x^k, \boldsymbol{Y}^0, \dots, \boldsymbol{Y}^k, \boldsymbol{Z}^0, \dots, \boldsymbol{Z}^k, h^0, \dots, h^k, \boldsymbol{P}^0, \dots, \boldsymbol{P}^{k-1}\}, \quad \mathcal{F}_k^+ = \sigma(\mathcal{F}_k \cup \{\boldsymbol{P}^k\}).$$

Define the projected functions on $\mathcal{B} := \prod_{\ell=1}^L \mathbb{R}^{r_\ell \times n_\ell}$ by

$$\tilde{f}_k(x, \boldsymbol{B}) := f(x, \boldsymbol{Y}^k + \boldsymbol{P}^k \boldsymbol{B}), \quad \tilde{g}_k(x, \boldsymbol{B}) := g(x, \boldsymbol{Y}^k + \boldsymbol{P}^k \boldsymbol{B}),$$

where $(\boldsymbol{P}^k \boldsymbol{B})_\ell := P_\ell^k B_\ell$ and $(\boldsymbol{P}^{k\top} \boldsymbol{U})_\ell := (P_\ell^k)^\top U_\ell$. The stochastic oracles $\tilde{U}_x^{k+1} \in \mathbb{R}^{d_x}, \tilde{\boldsymbol{U}}_Y^{k+1} \in \mathcal{B}, \tilde{\boldsymbol{V}}^{k+1} \in \mathcal{B}$, and the stochastic linear operators $\tilde{\mathcal{H}}^{k+1} : \mathcal{B} \to \mathcal{B}, \tilde{\mathcal{J}}^{k+1} : \mathcal{B} \to \mathbb{R}^{d_x}$ are conditionally unbiased given $\mathcal{F}_k^+$:

$$\begin{aligned}
\mathbb{E}[\tilde{U}_x^{k+1} \mid \mathcal{F}_k^+] &= \nabla_x \tilde{f}_k(x^k, 0), \\
\mathbb{E}[\tilde{\boldsymbol{U}}_Y^{k+1} \mid \mathcal{F}_k^+] &= \nabla_{\boldsymbol{B}} \tilde{f}_k(x^k, \boldsymbol{0}), \\
\mathbb{E}[\tilde{\boldsymbol{V}}^{k+1} \mid \mathcal{F}_k^+] &= \nabla_{\boldsymbol{B}} \tilde{g}_k(x^k, \boldsymbol{0}), \\
\mathbb{E}[\tilde{\mathcal{H}}^{k+1}[W] \mid \mathcal{F}_k^+] &= \nabla^2_{\boldsymbol{B}\boldsymbol{B}} \tilde{g}_k(x^k, \boldsymbol{0})[W], \quad \forall W \in \mathcal{B}, \\
\mathbb{E}[\tilde{\mathcal{J}}^{k+1}[W] \mid \mathcal{F}_k^+] &= \nabla^2_{x\boldsymbol{B}} \tilde{g}_k(x^k, \boldsymbol{0})[W], \quad \forall W \in \mathcal{B}.
\end{aligned}$$

Moreover, there exist constants $\tilde{\sigma}_{f,1}, \tilde{\sigma}_{f,2}, \tilde{\sigma}_{g,1}, \tilde{\sigma}_{g,2} \geq 0$ such that

$$\mathbb{E}[\|\tilde{U}_x^{k+1} - \nabla_x \tilde{f}_k(x^k, 0)\|^2 \mid \mathcal{F}_k^+] \leq \tilde{\sigma}_{f,1}^2,$$
$$\mathbb{E}[\|\tilde{\boldsymbol{U}}_Y^{k+1} - \nabla_{\boldsymbol{B}} \tilde{f}_k(x^k, \mathbf{0})\|^2 \mid \mathcal{F}_k^+] \leq \tilde{\sigma}_{f,2}^2,$$
$$\mathbb{E}[\|\tilde{\boldsymbol{V}}^{k+1} - \nabla_{\boldsymbol{B}} \tilde{g}_k(x^k, \mathbf{0})\|^2 \mid \mathcal{F}_k^+] \leq \tilde{\sigma}_{g,1}^2,$$
$$\mathbb{E}[\|\tilde{\mathcal{H}}^{k+1}[W] - \nabla^2_{\boldsymbol{BB}} \tilde{g}_k(x^k, \mathbf{0})[W]\|^2 \mid \mathcal{F}_k^+] \leq \tilde{\sigma}_{g,2}^2 \|W\|^2, \quad \forall W \in \mathcal{B},$$
$$\mathbb{E}[\|\tilde{\mathcal{J}}^{k+1}[W] - \nabla^2_{x\boldsymbol{B}} \tilde{g}_k(x^k, \mathbf{0})[W]\|^2 \mid \mathcal{F}_k^+] \leq \tilde{\sigma}_{g,2}^2 \|W\|^2, \quad \forall W \in \mathcal{B}.$$

In the algorithm, the layerwise probes $\{u_\ell^{k+1}, \xi_\ell^{k+1}\}_{\ell=1}^L$ are sampled independently of $\mathcal{F}_k^+$ and independently of the random oracle objects above. Define $\mathcal{F}_k^{++} := \sigma\big(\mathcal{F}_k^+, \{u_\ell^{k+1}, \xi_\ell^{k+1}\}_{\ell=1}^L\big)$. Conditioned on $\mathcal{F}_k^+$, the random objects $\tilde{U}_x^{k+1}, \tilde{\boldsymbol{U}}_Y^{k+1}, \tilde{\boldsymbol{V}}^{k+1}, \tilde{\mathcal{J}}^{k+1}, \tilde{\mathcal{H}}^{k+1}$ are mutually independent. The operators $\tilde{\mathcal{H}}^{k+1}$ and $\tilde{\mathcal{J}}^{k+1}$ are sampled once at iteration k and may be queried multiple times within the same iteration; no additional conditional independence is assumed between distinct queries to the same operator realization. Moreover, the displayed first- and second-moment relations for $\tilde{\mathcal{H}}^{k+1}$ and $\tilde{\mathcal{J}}^{k+1}$ extend to every $\mathcal{F}_k^{++}$-measurable random direction $W \in \mathcal{B}$ with finite conditional second moment.

Assumption 6 (Random orthogonal projection)**.** *Conditional on $\mathcal{F}_k$, the matrices $\{P_\ell^k\}_{\ell=1}^L$ are independent and satisfy*

$$P_\ell^k = \sqrt{\frac{m_\ell}{r_\ell}}\, O_{\ell,r_\ell}^k,$$

where O_{ℓ,r_ℓ}^k is formed by the first r_ℓ columns of a Haar-uniform $O_\ell^k \in O(m_\ell)$. Consequently, $(P_\ell^k)^\top P_\ell^k = (m_\ell/r_\ell) I_{r_\ell}$ and $\mathbb{E}[P_\ell^k (P_\ell^k)^\top \mid \mathcal{F}_k] = I_{m_\ell}$. We assume $r_\ell \geq 2$ for all ℓ.

C.2 Proof roadmap

This appendix proves the two claims stated in Section 3: the moment-preservation lemma, which justifies the bias-corrected projected directions, and the main convergence theorem, which gives the $\mathcal{O}(\varepsilon^{-2})$ sample complexity with projection-ratio dependent constants. For the convergence part, we use the Lyapunov quantity

$$V_k := \|x_+^k - x^k\|^2 + \|h^k - \nabla\Phi(x^k)\|^2,$$

where $x_+^k := x^k - h^k$. It is enough to prove $\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E} V_k \leq C_0/(K\alpha) + C_1\alpha$, because $\|\nabla\Phi(x^k)\|^2 \leq 2V_k$. The proof is organized as follows.

- **Step 1 (bias-corrected moment preservation).** Lemmas 2–5 establish the technical bridge behind Lemma 1. The standard implicit-function identities are first extended to the product Frobenius space. The main new point is the auxiliary Hessian action: Lemma 3 identifies the projection distortion in the self-layer blocks, Lemma 4 shows that the Rademacher bi-probe does not introduce cross-layer mean errors, and Lemma 5 proves that the final **BROS** directions satisfy

  $$\mathbb{E}[\widehat{\boldsymbol{V}}^{k+1} \mid \mathcal{F}_k] = D_Y^k, \quad \mathbb{E}[\widehat{\boldsymbol{S}}^{k+1} \mid \mathcal{F}_k] = D_Z^k, \quad \mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] = D_x^k,$$

  with conditional second moments controlled by the tracking errors. The leading projection-ratio dependence is tracked through $\omega_P = \max_\ell m_\ell/r_\ell$, while fixed dimension- and model-dependent constants are absorbed into the generic constants.
- **Step 2 (tracking the lower and auxiliary solutions).** With the centered directions from Step 1, Lemma 6 proves that the iterates $\boldsymbol{Y}^k$ and $\boldsymbol{Z}^k$ track the moving targets $\boldsymbol{Y}^*(x^k)$ and $\boldsymbol{Z}^*(x^k)$. In weighted-sum form, the tracking errors are bounded by the primal progress term $\sum_k \alpha\mathbb{E}\|x_+^k - x^k\|^2$, initialization constants, and stochastic terms of order $K\alpha^2$.
- **Step 3 (hypergradient tracking).** Lemma 7 converts the lower/auxiliary tracking errors from Step 2 into a bound on the bias between the centered estimator $\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k]$ and the true hypergradient $\nabla\Phi(x^k)$. Lemma 8 controls the variance of $\widehat{W}^{k+1}$, and Lemma 9 turns these estimates into a recursion for the moving-average error $S_h := \sum_{k=0}^{K-1} \alpha\mathbb{E}\|h^k - \nabla\Phi(x^k)\|^2$:

  $$S_h \leq A_0 + \rho_0 S_x + A_1 K\alpha^2 + A_2\alpha,$$

where $S_x := \sum_{k=0}^{K-1} \alpha \mathbb{E}\|x_+^k - x^k\|^2$ and ρ_0 is defined in (79).

- **Step 4 (closing the descent recursion).** Lemma 10 gives the descent relation $S_x \le 4(\Phi(x^0) - \Phi^*) + 2S_h$. Substituting this into the Step 3 recursion and choosing the coupled stepsizes so that ρ_0 and the remaining α-dependent coefficient are small allows the S_h term to be absorbed. This yields

$$S_h + S_x \le \mathcal{O}(1) + \mathcal{O}(K\alpha^2).$$

Dividing by $K\alpha$ gives the formal bound in Theorem 2. After absorbing fixed dimension- and model-dependent quantities into generic constants, the displayed projection-ratio dependence in the main-text form of Theorem 1 is expressed through ω_P: the transient and stochastic-error terms scale as $1+\omega_P^2$ and $1+\omega_P^4$, respectively, and the admissible stepsize is $\alpha \le \bar{\alpha}/(1+\omega_P^3)$. Choosing $\alpha = \Theta(K^{-1/2})$ then gives $\mathbb{E}\|\nabla\Phi(x^R)\|^2 = \mathcal{O}((1+\omega_P^4)K^{-1/2})$, hence $K = \mathcal{O}((1+\omega_P^4)^2\varepsilon^{-2})$ for an ε-stationary point.

C.3 Supporting lemmas

Lemma 2 (Product-space smoothness consequences)**.** *Suppose that* (A1) *holds. Let $\boldsymbol{Y}^*(x)$ be the unique minimizer of $\boldsymbol{Y} \mapsto g(x, \boldsymbol{Y})$ and define*

$$\boldsymbol{Z}^*(x) := \mathcal{H}(x, \boldsymbol{Y}^*(x))^{-1}[\nabla_{\boldsymbol{Y}} f(x, \boldsymbol{Y}^*(x))] \in \mathcal{Y}. \tag{14}$$

Then the following statements hold.

1. ***Implicit hypergradient formula.*** *The bilevel objective $\Phi(x) = f(x, \boldsymbol{Y}^*(x))$ is differentiable and*

$$\nabla\Phi(x) = \nabla_x f(x, \boldsymbol{Y}^*(x)) - \mathcal{J}(x, \boldsymbol{Y}^*(x))[\boldsymbol{Z}^*(x)]. \tag{15}$$

2. ***Lipschitz continuity of $\boldsymbol{Y}^*(\cdot)$.*** *The mapping $\boldsymbol{Y}^*(\cdot)$ is Lipschitz continuous with*

$$\|\boldsymbol{Y}^*(x) - \boldsymbol{Y}^*(\tilde{x})\| \le L_{Y^*}\|x - \tilde{x}\|, \quad L_{Y^*} := \frac{L_{g,1}}{\mu_g} \le \kappa. \tag{16}$$

3. ***Lipschitz continuity of $\boldsymbol{Z}^*(\cdot)$.*** *The mapping $\boldsymbol{Z}^*(\cdot)$ is Lipschitz continuous with*

$$\|\boldsymbol{Z}^*(x) - \boldsymbol{Z}^*(\tilde{x})\| \le L_{Z^*}\|x - \tilde{x}\|, \quad L_{Z^*} := \sqrt{1 + L_{Y^*}^2}\left(\frac{L_{f,1}}{\mu_g} + \frac{L_{f,0}L_{g,2}}{\mu_g^2}\right). \tag{17}$$

In particular, if $\kappa \ge 1$ then $L_{Z^*} \le 2\sqrt{2}\,\kappa^3$.

4. ***Lipschitz continuity of $\nabla\Phi$.*** *The gradient $\nabla\Phi$ is Lipschitz continuous with*

$$\|\nabla\Phi(x) - \nabla\Phi(\tilde{x})\| \le L_{\nabla\Phi}\|x - \tilde{x}\|, \tag{18}$$

where one may take

$$L_{\nabla\Phi} := \sqrt{1 + L_{Y^*}^2}\left(L_{f,1} + \frac{L_{f,0}L_{g,2}}{\mu_g} + \frac{L_{g,1}L_{f,1}}{\mu_g} + \frac{L_{g,1}L_{f,0}L_{g,2}}{\mu_g^2}\right). \tag{19}$$

5. ***Uniform bound on $\|\boldsymbol{Z}^*(x)\|$.*** *For all x,*

$$\|\boldsymbol{Z}^*(x)\| \le \frac{L_{f,0}}{\mu_g}. \tag{20}$$

Proof. We endow $\mathbb{R}^{d_x} \times \mathcal{Y}$ with the product norm $\|(x, \boldsymbol{Y})\| := \sqrt{\|x\|^2 + \|\boldsymbol{Y}\|^2}$.

Step 1: implicit differentiation and (15)**.** Fix x and write $\boldsymbol{Y}^* := \boldsymbol{Y}^*(x)$. Define

$$F(x, \boldsymbol{Y}) := \nabla_{\boldsymbol{Y}} g(x, \boldsymbol{Y}) \in \mathcal{Y}.$$

By Assumption 1, g is twice continuously differentiable, so F is continuously differentiable with

$$\nabla_{\boldsymbol{Y}} F(x, \boldsymbol{Y}) = \mathcal{H}(x, \boldsymbol{Y}), \quad \nabla_x F(x, \boldsymbol{Y}) = \nabla^2_{\boldsymbol{Y}x} g(x, \boldsymbol{Y}).$$

Since $\boldsymbol{Y} \mapsto g(x, \boldsymbol{Y})$ is μ_g-strongly convex in the product Frobenius norm, its Hessian is uniformly positive definite:

$$\langle \boldsymbol{W}, \mathcal{H}(x, \boldsymbol{Y})[\boldsymbol{W}] \rangle \geq \mu_g \|\boldsymbol{W}\|^2, \quad \forall \boldsymbol{W} \in \mathcal{Y},\ \forall (x, \boldsymbol{Y}). \tag{21}$$

Hence $\mathcal{H}(x, \boldsymbol{Y}^*)$ is invertible. Applying the implicit function theorem to $F(x, \boldsymbol{Y}) = 0$ at $(x, \boldsymbol{Y}^*)$ shows that $\boldsymbol{Y}^*(\cdot)$ is continuously differentiable locally, and differentiating $F(x, \boldsymbol{Y}^*(x)) \equiv 0$ in direction $u \in \mathbb{R}^{d_x}$ gives

$$0 = \nabla^2_{\boldsymbol{Y}x} g(x, \boldsymbol{Y}^*)[u] + \mathcal{H}(x, \boldsymbol{Y}^*)[D\boldsymbol{Y}^*(x)[u]],$$

so

$$D\boldsymbol{Y}^*(x)[u] = -\mathcal{H}(x, \boldsymbol{Y}^*)^{-1}[\nabla^2_{\boldsymbol{Y}x} g(x, \boldsymbol{Y}^*)[u]]. \tag{22}$$

Now compute the directional derivative of $\Phi(x) = f(x, \boldsymbol{Y}^*(x))$:

$$D\Phi(x)[u] = \langle \nabla_x f(x, \boldsymbol{Y}^*), u \rangle + \langle \nabla_{\boldsymbol{Y}} f(x, \boldsymbol{Y}^*), D\boldsymbol{Y}^*(x)[u] \rangle.$$

Substituting (22) and using that $\mathcal{H}(x, \boldsymbol{Y}^*)$ is self-adjoint on $\mathcal{Y}$ (hence so is its inverse) yields

$$D\Phi(x)[u] = \langle \nabla_x f(x, \boldsymbol{Y}^*), u \rangle - \Big\langle \mathcal{H}(x, \boldsymbol{Y}^*)^{-1}[\nabla_{\boldsymbol{Y}} f(x, \boldsymbol{Y}^*)], \nabla^2_{\boldsymbol{Y}x} g(x, \boldsymbol{Y}^*)[u] \Big\rangle.$$

We define $\mathcal{J}(x, \boldsymbol{Y}) : \mathcal{Y} \to \mathbb{R}^{d_x}$ as the adjoint of $u \mapsto \nabla^2_{\boldsymbol{Y}x} g(x, \boldsymbol{Y})[u]$, i.e., for all $u \in \mathbb{R}^{d_x}$ and $\boldsymbol{W} \in \mathcal{Y}$,

$$\langle \nabla^2_{\boldsymbol{Y}x} g(x, \boldsymbol{Y})[u], \boldsymbol{W} \rangle = \langle u, \mathcal{J}(x, \boldsymbol{Y})[\boldsymbol{W}] \rangle. \tag{23}$$

Recalling the definition (14), we obtain

$$D\Phi(x)[u] = \langle \nabla_x f(x, \boldsymbol{Y}^*) - \mathcal{J}(x, \boldsymbol{Y}^*)[\boldsymbol{Z}^*(x)], u \rangle,$$

which proves (15).

Step 2: Lipschitz continuity of $\boldsymbol{Y}^*(\cdot)$. Fix $x, \tilde{x} \in \mathbb{R}^{d_x}$. Using the optimality conditions

$$\nabla_{\boldsymbol{Y}} g(x, \boldsymbol{Y}^*(x)) = 0, \quad \nabla_{\boldsymbol{Y}} g(\tilde{x}, \boldsymbol{Y}^*(\tilde{x})) = 0,$$

and the strong monotonicity of $\boldsymbol{Y} \mapsto \nabla_{\boldsymbol{Y}} g(x, \boldsymbol{Y})$, we get

$$\begin{aligned} \mu_g \|\boldsymbol{Y}^*(x) - \boldsymbol{Y}^*(\tilde{x})\| &\leq \|\nabla_{\boldsymbol{Y}} g(x, \boldsymbol{Y}^*(\tilde{x})) - \nabla_{\boldsymbol{Y}} g(x, \boldsymbol{Y}^*(x))\| \\ &= \|\nabla_{\boldsymbol{Y}} g(\tilde{x}, \boldsymbol{Y}^*(\tilde{x})) - \nabla_{\boldsymbol{Y}} g(x, \boldsymbol{Y}^*(\tilde{x}))\|. \end{aligned}$$

By the $L_{g,1}$-Lipschitz continuity of ∇g in Assumption 1, the last term is at most $L_{g,1}\|x - \tilde{x}\|$, which proves (16).

Step 3: Lipschitz continuity and the bound of $\boldsymbol{Z}^*(\cdot)$. Define

$$A(x) := \mathcal{H}(x, \boldsymbol{Y}^*(x)), \quad b(x) := \nabla_{\boldsymbol{Y}} f(x, \boldsymbol{Y}^*(x)), \quad \boldsymbol{Z}^*(x) = A(x)^{-1}[b(x)].$$

From (21), $\|A(x)^{-1}\| \leq 1/\mu_g$. Assumption 1 gives $\|b(x)\| \leq L_{f,0}$, so $\|\boldsymbol{Z}^*(x)\| \leq \frac{L_{f,0}}{\mu_g}$.

Next, by Assumption 1 and (16),

$$\|b(x) - b(\tilde{x})\| \leq L_{f,1}\|(x, \boldsymbol{Y}^*(x)) - (\tilde{x}, \boldsymbol{Y}^*(\tilde{x}))\| \leq L_{f,1}\sqrt{1 + L^2_{Y^*}}\|x - \tilde{x}\|. \tag{24}$$

Similarly, since $\nabla^2 g$ is $L_{g,2}$-Lipschitz,

$$\|A(x) - A(\tilde{x})\| \leq L_{g,2}\|(x, \boldsymbol{Y}^*(x)) - (\tilde{x}, \boldsymbol{Y}^*(\tilde{x}))\| \leq L_{g,2}\sqrt{1 + L^2_{Y^*}}\|x - \tilde{x}\|. \tag{25}$$

Using the resolvent identity $A(x)^{-1} - A(\tilde{x})^{-1} = A(x)^{-1}(A(\tilde{x}) - A(x))A(\tilde{x})^{-1}$, we have

$$\|A(x)^{-1} - A(\tilde{x})^{-1}\| \leq \frac{1}{\mu_g^2}\|A(x) - A(\tilde{x})\|. \tag{26}$$

Decomposing

$$\boldsymbol{Z}^*(x) - \boldsymbol{Z}^*(\tilde{x}) = A(x)^{-1}[b(x) - b(\tilde{x})] + (A(x)^{-1} - A(\tilde{x})^{-1})[b(\tilde{x})]$$

and combining (24)–(26) with $\|b(\tilde{x})\| \leq L_{f,0}$ yields (17).

Step 4: Lipschitz continuity of $\nabla\Phi$. From (15),

$$\nabla\Phi(x) = \nabla_x f(x, \boldsymbol{Y}^*(x)) - \mathcal{J}(x, \boldsymbol{Y}^*(x))[\boldsymbol{Z}^*(x)].$$

Because ∇g is $L_{g,1}$-Lipschitz, the full Hessian of g has operator norm at most $L_{g,1}$; in particular,

$$\|\mathcal{J}(x, \boldsymbol{Y})\| \le L_{g,1}, \quad \forall (x, \boldsymbol{Y}) \in \mathbb{R}^{d_x} \times \mathcal{Y}. \tag{27}$$

Likewise, the $L_{g,2}$-Lipschitz continuity of $\nabla^2 g$ implies

$$\|\mathcal{J}(x, \boldsymbol{Y}) - \mathcal{J}(\tilde{x}, \tilde{\boldsymbol{Y}})\| \le L_{g,2}\|(x, \boldsymbol{Y}) - (\tilde{x}, \tilde{\boldsymbol{Y}})\|. \tag{28}$$

Therefore,

$$\begin{aligned}\|\nabla\Phi(x) - \nabla\Phi(\tilde{x})\| \le{}& \|\nabla_x f(x, \boldsymbol{Y}^*(x)) - \nabla_x f(\tilde{x}, \boldsymbol{Y}^*(\tilde{x}))\| \\ &+ \|\mathcal{J}(x, \boldsymbol{Y}^*(x))[\boldsymbol{Z}^*(x)] - \mathcal{J}(\tilde{x}, \boldsymbol{Y}^*(\tilde{x}))[\boldsymbol{Z}^*(\tilde{x})]\|.\end{aligned}$$

The first term is bounded by $L_{f,1}\sqrt{1 + L_{Y^*}^2}\|x - \tilde{x}\|$. For the second term, add and subtract $\mathcal{J}(\tilde{x}, \boldsymbol{Y}^*(\tilde{x}))[\boldsymbol{Z}^*(x)]$:

$$\begin{aligned}\|\mathcal{J}(x, \boldsymbol{Y}^*(x))[\boldsymbol{Z}^*(x)] - \mathcal{J}(\tilde{x}, \boldsymbol{Y}^*(\tilde{x}))[\boldsymbol{Z}^*(\tilde{x})]\| \le{}& \|\mathcal{J}(x, \boldsymbol{Y}^*(x)) - \mathcal{J}(\tilde{x}, \boldsymbol{Y}^*(\tilde{x}))\|\,\|\boldsymbol{Z}^*(x)\| \\ &+ \|\mathcal{J}(\tilde{x}, \boldsymbol{Y}^*(\tilde{x}))\|\,\|\boldsymbol{Z}^*(x) - \boldsymbol{Z}^*(\tilde{x})\|.\end{aligned}$$

Using (28), (27), and (17) yields (18) with the constant (19). □

Lemma 3 (Layerwise projection identities and induced derivatives)**.** *For each layer $\ell \in \{1, \dots, L\}$, let*

$$P_\ell := \sqrt{\frac{m_\ell}{r_\ell}}\, O_{\ell, r_\ell} \in \mathbb{R}^{m_\ell \times r_\ell}, \quad Q_\ell := P_\ell P_\ell^\top,$$

where O_{ℓ,r_ℓ} contains the first r_ℓ columns of a Haar-uniform matrix in $O(m_\ell)$. Assume the projectors are independent across layers. Then:

(i) For every ℓ, we have $P_\ell^\top P_\ell = \frac{m_\ell}{r_\ell} I_{r_\ell}$ and $\mathbb{E}[Q_\ell] = I_{m_\ell}$.

(ii) For every layer ℓ, there exist constants

$$a_\ell = \frac{m_\ell(m_\ell r_\ell + m_\ell - 2)}{r_\ell(m_\ell - 1)(m_\ell + 2)}, \quad b_\ell = \frac{m_\ell(m_\ell - r_\ell)}{r_\ell(m_\ell - 1)(m_\ell + 2)} \tag{29}$$

such that for any deterministic $A \in \mathbb{R}^{m_\ell \times m_\ell}$,

$$\mathbb{E}[Q_\ell A Q_\ell] = (a_\ell - b_\ell)A + b_\ell A^\top + b_\ell \operatorname{tr}(A) I_{m_\ell}, \tag{30}$$

and

$$a_\ell - 2b_\ell = \frac{m_\ell(r_\ell - 1)}{r_\ell(m_\ell - 1)} > 0. \tag{31}$$

(iii) For any pair $\ell \neq t$ and any deterministic $A \in \mathbb{R}^{m_\ell \times m_t}$,

$$\mathbb{E}[Q_\ell A Q_t] = A. \tag{32}$$

(iv) Fix an iteration k and define

$$\tilde{f}_k(x, \boldsymbol{B}) := f(x, \boldsymbol{Y}^k + \boldsymbol{P}^k \boldsymbol{B}), \quad \tilde{g}_k(x, \boldsymbol{B}) := g(x, \boldsymbol{Y}^k + \boldsymbol{P}^k \boldsymbol{B}).$$

Then at $\boldsymbol{B} = \boldsymbol{0}$,

$$\nabla_{\boldsymbol{B}} \tilde{f}_k(x, \boldsymbol{0}) = \boldsymbol{P}^{k\top} \nabla_{\boldsymbol{Y}} f(x, \boldsymbol{Y}^k), \quad \nabla_{\boldsymbol{B}} \tilde{g}_k(x, \boldsymbol{0}) = \boldsymbol{P}^{k\top} \nabla_{\boldsymbol{Y}} g(x, \boldsymbol{Y}^k),$$

and for every $\boldsymbol{W} \in \mathcal{B}$,

$$\begin{aligned}\nabla^2_{\boldsymbol{B}\boldsymbol{B}} \tilde{g}_k(x, \boldsymbol{0})[\boldsymbol{W}] &= \boldsymbol{P}^{k\top} \mathcal{H}(x, \boldsymbol{Y}^k)[\boldsymbol{P}^k \boldsymbol{W}], \\ \nabla^2_{x\boldsymbol{B}} \tilde{g}_k(x, \boldsymbol{0})[\boldsymbol{W}] &= \mathcal{J}(x, \boldsymbol{Y}^k)[\boldsymbol{P}^k \boldsymbol{W}].\end{aligned}$$

Proof. **Proof of (i).** Fix a layer ℓ. Since the columns of O_{ℓ,r_ℓ} are orthonormal,

$$P_\ell^\top P_\ell = \frac{m_\ell}{r_\ell} O_{\ell,r_\ell}^\top O_{\ell,r_\ell} = \frac{m_\ell}{r_\ell} I_{r_\ell}.$$

To prove $\mathbb{E}[Q_\ell] = I_{m_\ell}$, let $V \in O(m_\ell)$ be deterministic. By left-invariance of Haar measure, VO_{ℓ,r_ℓ} has the same distribution as O_{ℓ,r_ℓ}, hence

$$V\,\mathbb{E}[Q_\ell]\,V^\top = \mathbb{E}[VQ_\ell V^\top] = \mathbb{E}[Q_\ell].$$

Therefore $\mathbb{E}[Q_\ell]$ commutes with every orthogonal matrix and must equal cI_{m_ℓ} for some scalar c. Taking traces and using $P_\ell^\top P_\ell = (m_\ell/r_\ell)I_{r_\ell}$ gives

$$cm_\ell = \operatorname{tr}(\mathbb{E}[Q_\ell]) = \mathbb{E}[\operatorname{tr}(P_\ell^\top P_\ell)] = \operatorname{tr}\left(\frac{m_\ell}{r_\ell} I_{r_\ell}\right) = m_\ell,$$

so $c = 1$.

Proof of (ii). Again fix ℓ and abbreviate

$$m := m_\ell, \quad r := r_\ell, \quad P := P_\ell, \quad Q := Q_\ell.$$

Let $Q_0 := O_r O_r^\top$, so $Q = (m/r)Q_0$. By conjugation invariance of Q_0, the fourth-order tensor

$$T_{iabj} := \mathbb{E}[(Q_0)_{ia}(Q_0)_{bj}]$$

is $O(m)$-invariant and hence has the form

$$\mathbb{E}[(Q_0)_{ia}(Q_0)_{bj}] = c_1\delta_{ia}\delta_{bj} + c_2\delta_{ib}\delta_{aj} + c_3\delta_{ij}\delta_{ab}.$$

To identify the coefficients, note first that for a Haar row vector $u \in \mathbb{S}^{m-1}$,

$$(Q_0)_{11} = \sum_{j=1}^{r} u_j^2,$$

so the standard sphere moments give $\mathbb{E}[(Q_0)_{11}^2] = \frac{r(r+2)}{m(m+2)}$. Using $\operatorname{tr}(Q_0^2) = \operatorname{tr}(Q_0) = r$ together with rotational invariance yields $\mathbb{E}[(Q_0)_{12}^2] = \frac{r(m-r)}{m(m-1)(m+2)}$. Evaluating the invariant-tensor form at $(i,a,b,j) = (1,2,1,2)$, $(1,2,2,1)$, and $(1,1,1,1)$ shows that

$$c_2 = c_3 = \frac{r(m-r)}{m(m-1)(m+2)}, \quad c_1 = \frac{r(r(m+1)-2)}{m(m-1)(m+2)}.$$

Hence for any $A \in \mathbb{R}^{m\times m}$,

$$\mathbb{E}[Q_0 A Q_0] = c_1 A + c_2 A^\top + c_2 \operatorname{tr}(A) I_m.$$

Multiplying by $(m/r)^2$ and simplifying gives

$$\mathbb{E}[QAQ] = (a_\ell - b_\ell)A + b_\ell A^\top + b_\ell \operatorname{tr}(A) I_{m_\ell},$$

with

$$a_\ell = \frac{m_\ell(m_\ell r_\ell + m_\ell - 2)}{r_\ell(m_\ell - 1)(m_\ell + 2)}, \quad b_\ell = \frac{m_\ell(m_\ell - r_\ell)}{r_\ell(m_\ell - 1)(m_\ell + 2)}.$$

The formula $a_\ell - 2b_\ell = \frac{m_\ell(r_\ell - 1)}{r_\ell(m_\ell - 1)} > 0$ follows by direct algebra.

Proof of (iii). For $\ell \neq t$ and deterministic $A \in \mathbb{R}^{m_\ell \times m_t}$, independence across layers gives

$$\mathbb{E}[Q_\ell A Q_t] = \mathbb{E}[Q_\ell]\, A\, \mathbb{E}[Q_t] = A,$$

where we used (i).

Proof of (iv). Fix k and define $\Psi(\boldsymbol{B}) := \boldsymbol{Y}^k + \boldsymbol{P}^k \boldsymbol{B}$. For any $\boldsymbol{H} \in \mathcal{B}$,

$$\begin{aligned}\langle \nabla_{\boldsymbol{B}} \tilde{g}_k(x, \boldsymbol{B}), \boldsymbol{H} \rangle &= \frac{\mathrm{d}}{\mathrm{d}t} g\big(x, \Psi(\boldsymbol{B} + t\boldsymbol{H})\big)\Big|_{t=0} \\ &= \langle \nabla_{\boldsymbol{Y}} g(x, \Psi(\boldsymbol{B})), \boldsymbol{P}^k \boldsymbol{H} \rangle = \langle \boldsymbol{P}^{k\top} \nabla_{\boldsymbol{Y}} g(x, \Psi(\boldsymbol{B})), \boldsymbol{H} \rangle,\end{aligned}$$

so

$$\nabla_{\boldsymbol{B}} \tilde{g}_k(x, \boldsymbol{B}) = \boldsymbol{P}^{k\top} \nabla_{\boldsymbol{Y}} g(x, \Psi(\boldsymbol{B})).$$

Evaluating at $\boldsymbol{B} = \boldsymbol{0}$ proves the formula for $\tilde{g}_k$; the proof for $\tilde{f}_k$ is identical.

For the Hessian, the directional derivative satisfies

$$\nabla^2_{\boldsymbol{BB}} \tilde{g}_k(x, \boldsymbol{B})[\boldsymbol{W}] = \frac{\mathrm{d}}{\mathrm{d}t} \nabla_{\boldsymbol{B}} \tilde{g}_k(x, \boldsymbol{B} + t\boldsymbol{W})\Big|_{t=0} = \boldsymbol{P}^{k\top} \mathcal{H}(x, \Psi(\boldsymbol{B}))[\boldsymbol{P}^k \boldsymbol{W}],$$

which gives the first identity at $\boldsymbol{B} = \boldsymbol{0}$. Similarly,

$$\nabla^2_{x\boldsymbol{B}} \tilde{g}_k(x, \boldsymbol{B})[\boldsymbol{W}] = \frac{\mathrm{d}}{\mathrm{d}t} \nabla_x g\big(x, \Psi(\boldsymbol{B} + t\boldsymbol{W})\big)\Big|_{t=0} = \mathcal{J}(x, \Psi(\boldsymbol{B}))[\boldsymbol{P}^k \boldsymbol{W}],$$

and evaluating at $\boldsymbol{B} = \boldsymbol{0}$ yields the claim. □

Lemma 4 (Cross-layer cancellation in the global probe)**.** *Fix an iteration k and a layer $t \in \{1, \dots, L\}$. Let $E_j : \mathbb{R}^{r_j \times n_j} \to \mathcal{B}$ denote the block-insertion operator into block j, define*

$$\Delta B_j^{k+1} := \xi_j^{k+1} (u_j^{k+1})^\top, \quad \Delta \boldsymbol{B}^{k+1} := (\Delta B_1^{k+1}, \dots, \Delta B_L^{k+1}),$$

and let $\boldsymbol{M}^{k+1} := \tilde{\mathcal{H}}^{k+1}[\Delta \boldsymbol{B}^{k+1}]$. Write the t-th output block as

$$M_t^{k+1} = M_t^{\mathrm{self},k+1} + M_t^{\mathrm{cross},k+1},$$

where $M_t^{\mathrm{self},k+1} := \big(\tilde{\mathcal{H}}^{k+1}[E_t(\Delta B_t^{k+1})]\big)_t$ and $M_t^{\mathrm{cross},k+1} := \sum_{j \neq t} \big(\tilde{\mathcal{H}}^{k+1}[E_j(\Delta B_j^{k+1})]\big)_t$. Let

$$\tilde{Z}_t^k := (P_t^k)^\top Z_t^k, \quad v_t^{\mathrm{self},k+1} := (M_t^{\mathrm{self},k+1})^\top \xi_t^{k+1}, \quad v_t^{\mathrm{cross},k+1} := (M_t^{\mathrm{cross},k+1})^\top \xi_t^{k+1},$$

$$w_t^{\mathrm{self},k+1} := (M_t^{\mathrm{self},k+1})^\top (\tilde{Z}_t^k u_t^{k+1}), \quad w_t^{\mathrm{cross},k+1} := (M_t^{\mathrm{cross},k+1})^\top (\tilde{Z}_t^k u_t^{k+1}),$$

and define

$$\mathcal{G}_{k,t} := \sigma\Big(\mathcal{F}_k^+, \tilde{\mathcal{H}}^{k+1}, \boldsymbol{Z}^k, \{u_j^{k+1}, \xi_j^{k+1}\}_{j \neq t}\Big).$$

Then

$$\mathbb{E}\Big[v_t^{\mathrm{cross},k+1} (u_t^{k+1})^\top \mid \mathcal{G}_{k,t}\Big] = 0, \tag{33a}$$

$$\mathbb{E}\Big[\xi_t^{k+1} (w_t^{\mathrm{cross},k+1})^\top \mid \mathcal{G}_{k,t}\Big] = 0. \tag{33b}$$

Consequently, the cross-layer contamination carried by the global probe contributes zero conditional mean to both

$$\widehat{C}_t^{k+1} := Z_t^k \big(v_t^{k+1} (u_t^{k+1})^\top\big), \quad \widehat{B}_{\sharp,t}^{k+1} := P_t^k \big(\xi_t^{k+1} (w_t^{k+1})^\top\big).$$

Proof. By construction, $M_t^{\mathrm{cross},k+1}$ depends only on the shared operator realization $\tilde{\mathcal{H}}^{k+1}$ and on the probes $\{u_j^{k+1}, \xi_j^{k+1}\}_{j \neq t}$, so it does not depend on u_t^{k+1}. Hence, conditioning first on $(\mathcal{G}_{k,t}, \xi_t^{k+1})$ and using the zero mean of the Rademacher vector u_t^{k+1} gives

$$\mathbb{E}\Big[v_t^{\mathrm{cross},k+1} (u_t^{k+1})^\top \mid \mathcal{G}_{k,t}, \xi_t^{k+1}\Big] = v_t^{\mathrm{cross},k+1}\, \mathbb{E}\big[(u_t^{k+1})^\top \mid \mathcal{G}_{k,t}, \xi_t^{k+1}\big] = 0.$$

Taking the tower expectation yields (33a).

Similarly, $M_t^{\mathrm{cross},k+1}$ and therefore $w_t^{\mathrm{cross},k+1}$ depend on u_t^{k+1} and on the off-layer probes, but not on ξ_t^{k+1}. Conditioning first on $(\mathcal{G}_{k,t}, u_t^{k+1})$ and using the zero mean of the Rademacher vector ξ_t^{k+1} gives

$$\mathbb{E}\Big[\xi_t^{k+1} (w_t^{\mathrm{cross},k+1})^\top \mid \mathcal{G}_{k,t}, u_t^{k+1}\Big] = \mathbb{E}\big[\xi_t^{k+1} \mid \mathcal{G}_{k,t}, u_t^{k+1}\big]\, (w_t^{\mathrm{cross},k+1})^\top = 0.$$

Taking the tower expectation yields (33b). The final claim follows by linearity from the decompositions

$$v_t^{k+1} = v_t^{\mathrm{self},k+1} + v_t^{\mathrm{cross},k+1}, \quad w_t^{k+1} = w_t^{\mathrm{self},k+1} + w_t^{\mathrm{cross},k+1}.$$

□

Lemma 5 (Unbiasedness and second moments of the projected stochastic directions)**.** *Assume* (A1)–(A3) *and the sampling conditions stated in the main text. Define* $\widehat{\boldsymbol{V}}^{k+1} := \boldsymbol{P}^k \tilde{\boldsymbol{V}}^{k+1} \in \mathcal{Y}$, $\widehat{W}^{k+1} := \tilde{U}_x^{k+1} - \tilde{\mathcal{J}}^{k+1}[\boldsymbol{P}^k{}^\top \boldsymbol{Z}^k] \in \mathbb{R}^{d_x}$ *and* $\widehat{\boldsymbol{S}}^{k+1} := \widehat{\mathcal{H}[\boldsymbol{Z}]}^{k+1} - \boldsymbol{P}^k \tilde{\boldsymbol{U}}_Y^{k+1} \in \mathcal{Y}$*, where* $\widehat{\mathcal{H}[\boldsymbol{Z}]}^{k+1}$ *is the assembled multilayer HVP estimator described in Appendix B.1. Let* $\omega_P := \max_{1\le \ell \le L} \frac{m_\ell}{r_\ell}$ *and* $\Lambda_Y := (\omega_P - 1) L_{g,1}^2$*. Then:*

(i) ***Conditional unbiasedness.***

$$\begin{aligned}
\mathbb{E}[\widehat{\boldsymbol{V}}^{k+1} \mid \mathcal{F}_k] &= \nabla_{\boldsymbol{Y}} g(x^k, \boldsymbol{Y}^k), \\
\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] &= \nabla_x f(x^k, \boldsymbol{Y}^k) - \mathcal{J}(x^k, \boldsymbol{Y}^k)[\boldsymbol{Z}^k], \\
\mathbb{E}[\widehat{\boldsymbol{S}}^{k+1} \mid \mathcal{F}_k] &= \mathcal{H}(x^k, \boldsymbol{Y}^k)[\boldsymbol{Z}^k] - \nabla_{\boldsymbol{Y}} f(x^k, \boldsymbol{Y}^k).
\end{aligned}$$

(ii) ***Conditional second moments.*** *Define*

$$\Delta_S^{k+1} := \widehat{\boldsymbol{S}}^{k+1} - \big(\mathcal{H}(x^k, \boldsymbol{Y}^k)[\boldsymbol{Z}^k] - \nabla_{\boldsymbol{Y}} f(x^k, \boldsymbol{Y}^k)\big).$$

There exist constants $\sigma_{S,0}, \sigma_{S,2}, \Lambda_S < \infty$ *such that*

$$\mathbb{E}[\|\widehat{\boldsymbol{V}}^{k+1} - \nabla_{\boldsymbol{Y}} g(x^k, \boldsymbol{Y}^k)\|^2 \mid \mathcal{F}_k] \le \omega_P \tilde{\sigma}_{g,1}^2 + \Lambda_Y \|\boldsymbol{Y}^k - \boldsymbol{Y}^*(x^k)\|^2, \tag{34a}$$

$$\mathbb{E}[\|\widehat{W}^{k+1} - \mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k]\|^2 \mid \mathcal{F}_k] \le \tilde{\sigma}_{f,1}^2 + (\tilde{\sigma}_{g,2}^2 + \Lambda_Y)\|\boldsymbol{Z}^k\|^2, \tag{34b}$$

$$\mathbb{E}\big[\|\Delta_S^{k+1}\|^2 \mid \mathcal{F}_k\big] \le \sigma_{S,0}^2 + \sigma_{S,2}^2 \|\boldsymbol{Y}^k - \boldsymbol{Y}^*(x^k)\|^2 + \Lambda_S \|\boldsymbol{Z}^k\|^2. \tag{34c}$$

Proof. We condition on $\mathcal{F}_k$ throughout and abbreviate

$$\boldsymbol{P} := \boldsymbol{P}^k, \quad P_\ell := P_\ell^k, \quad Q_\ell := P_\ell (P_\ell)^\top,$$

$$\boldsymbol{Y} := \boldsymbol{Y}^k, \quad \boldsymbol{Z} := \boldsymbol{Z}^k, \quad f_Y := \nabla_{\boldsymbol{Y}} f(x^k, \boldsymbol{Y}), \quad H := \mathcal{H}(x^k, \boldsymbol{Y}).$$

We also write

$$\mathcal{F}_k^+ = \sigma(\mathcal{F}_k \cup \{\boldsymbol{P}\}), \quad \mathcal{F}_k^{++} = \sigma\big(\mathcal{F}_k^+, \{u_\ell^{k+1}, \xi_\ell^{k+1}\}_{\ell=1}^L\big).$$

Part (i): unbiasedness of $\widehat{\boldsymbol{V}}^{k+1}$**.** By Assumption 5, $\mathbb{E}[\tilde{\boldsymbol{V}}^{k+1} \mid \mathcal{F}_k^+] = \nabla_{\boldsymbol{B}} \tilde{g}_k(x^k, \boldsymbol{0})$. Lemma 3(iv) gives $\nabla_{\boldsymbol{B}} \tilde{g}_k(x^k, \boldsymbol{0}) = \boldsymbol{P}^\top \nabla_{\boldsymbol{Y}} g(x^k, \boldsymbol{Y})$. Therefore,

$$\begin{aligned}
\mathbb{E}[\widehat{\boldsymbol{V}}^{k+1} \mid \mathcal{F}_k] &= \mathbb{E}\Big[\boldsymbol{P}\, \mathbb{E}[\tilde{\boldsymbol{V}}^{k+1} \mid \mathcal{F}_k^+] \mid \mathcal{F}_k\Big] \\
&= \mathbb{E}[\boldsymbol{P}\boldsymbol{P}^\top \mid \mathcal{F}_k] \nabla_{\boldsymbol{Y}} g(x^k, \boldsymbol{Y}) = \nabla_{\boldsymbol{Y}} g(x^k, \boldsymbol{Y}),
\end{aligned}$$

because each block projector satisfies $\mathbb{E}[Q_\ell \mid \mathcal{F}_k] = I_{m_\ell}$.

Part (i): unbiasedness of $\widehat{W}^{k+1}$**.** Again by Assumption 5,

$$\mathbb{E}[\tilde{U}_x^{k+1} \mid \mathcal{F}_k^+] = \nabla_x \tilde{f}_k(x^k, \boldsymbol{0}) = \nabla_x f(x^k, \boldsymbol{Y}).$$

Moreover, for every $W \in \mathcal{B}$,

$$\mathbb{E}[\tilde{\mathcal{J}}^{k+1}[W] \mid \mathcal{F}_k^+] = \nabla_{x\boldsymbol{B}}^2 \tilde{g}_k(x^k, \boldsymbol{0})[W].$$

Applying Lemma 3(iv) with $W = \boldsymbol{P}^\top \boldsymbol{Z}$ gives

$$\mathbb{E}[\tilde{\mathcal{J}}^{k+1}[\boldsymbol{P}^\top \boldsymbol{Z}] \mid \mathcal{F}_k^+] = \mathcal{J}(x^k, \boldsymbol{Y})[\boldsymbol{P}\boldsymbol{P}^\top \boldsymbol{Z}],$$

where $(\boldsymbol{P}\boldsymbol{P}^\top \boldsymbol{Z})_\ell = Q_\ell Z_\ell$. Taking conditional expectation with respect to $\mathcal{F}_k$ and using $\mathbb{E}[Q_\ell \mid \mathcal{F}_k] = I_{m_\ell}$ yields

$$\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] = \nabla_x f(x^k, \boldsymbol{Y}) - \mathcal{J}(x^k, \boldsymbol{Y})[\boldsymbol{Z}].$$

Part (i): unbiasedness of $\widehat{\boldsymbol{S}}^{k+1}$. For each pair of layers (ℓ, t), let $H_{\ell t} : \mathbb{R}^{m_t \times n_t} \to \mathbb{R}^{m_\ell \times n_\ell}$ denote the (ℓ, t) block of H, so that

$$(H[\boldsymbol{Z}])_\ell = \sum_{t=1}^{L} H_{\ell t}[Z_t].$$

For each (ℓ, t) and columns $i \in \{1, \dots, n_\ell\}$, $j \in \{1, \dots, n_t\}$, there exist matrices $H_{ij}^{\ell t} \in \mathbb{R}^{m_\ell \times m_t}$ such that for $U_t = [u_{t,1}, \dots, u_{t,n_t}]$,

$$(H_{\ell t}[U_t])_{:,i} = \sum_{j=1}^{n_t} H_{ij}^{\ell t} u_{t,j}.$$

Self-adjointness of H on $\mathcal{Y}$ implies $H_{ji}^{t\ell} = (H_{ij}^{\ell t})^\top$.

Step 1: deterministic expectation of the masked main HVPs. Condition on $\mathcal{F}_k^+$ and define the deterministic projected Hessian operator $\bar{\mathcal{H}} := \nabla^2_{\boldsymbol{BB}} \tilde{g}_k(x^k, \mathbf{0})$. For each input layer t, set $\tilde{Z}_t := (P_t)^\top Z_t$ and $\bar{\boldsymbol{A}}^{(t)} := \boldsymbol{P}\, \bar{\mathcal{H}}[E_t(\tilde{Z}_t)]$. By Lemma 3(iv),

$$(\bar{A}_\ell^{(t)})_{:,i} = \sum_{j=1}^{n_t} Q_\ell H_{ij}^{\ell t} Q_t z_{t,j}.$$

If $\ell \neq t$, Lemma 3(iii) gives

$$\mathbb{E}[\bar{A}_\ell^{(t)} \mid \mathcal{F}_k] = H_{\ell t}[Z_t].$$

For $\ell = t$, define the self-layer partial-transpose operator and trace matrix by

$$(H_{tt}^\sharp[Z_t])_{:,i} := \sum_{j=1}^{n_t} (H_{ij}^{tt})^\top z_{t,j}, \quad (T_t)_{ij} := \operatorname{tr}(H_{ij}^{tt}).$$

Applying Lemma 3(ii) blockwise to H_{ij}^{tt} yields

$$\mathbb{E}[\bar{A}_t^{(t)} \mid \mathcal{F}_k] = (a_t - b_t) H_{tt}[Z_t] + b_t H_{tt}^\sharp[Z_t] + b_t Z_t T_t. \tag{35}$$

Step 2: deterministic expectation of the global probe correction terms. Condition now on $\mathcal{F}_k^+$ and define

$$\bar{\boldsymbol{M}} := \bar{\mathcal{H}}[\Delta \boldsymbol{B}^{k+1}], \quad \Delta B_\ell^{k+1} := \xi_\ell^{k+1} (u_\ell^{k+1})^\top.$$

Fix a layer t. The same cancellation argument as in Lemma 4 applies to the deterministic operator $\bar{\mathcal{H}}$, so the cross-layer part of $\bar{M}_t$ contributes zero conditional mean to both

$$\widehat{C}_t^{k+1} = Z_t\big(v_t^{k+1}(u_t^{k+1})^\top\big), \quad \widehat{B}_{\sharp,t}^{k+1} = P_t\big(\xi_t^{k+1}(w_t^{k+1})^\top\big).$$

It is therefore enough to compute the expectation of the self part.

For the trace-matrix correction, define the projected trace matrix $(\tilde{T}_t)_{ij} := \operatorname{tr}\big((P_t)^\top H_{ij}^{tt} P_t\big)$. Writing out the self contribution columnwise and using $\mathbb{E}[\xi_t^{k+1}(\xi_t^{k+1})^\top] = I_{r_t}$, $\mathbb{E}[u_t^{k+1}(u_t^{k+1})^\top] = I_{n_t}$ gives

$$\mathbb{E}[\widehat{C}_t^{k+1} \mid \mathcal{F}_k^+] = Z_t \tilde{T}_t. \tag{36}$$

Taking expectation over P_t and using Lemma 3(i) gives:

$$\mathbb{E}[\widehat{C}_t^{k+1} \mid \mathcal{F}_k] = Z_t T_t. \tag{37}$$

For the partial-transpose correction, define the projected self-layer operator

$$(\bar{H}_{tt}^\sharp[W_t])_{:,i} := \sum_{j=1}^{n_t} \big((P_t)^\top H_{ij}^{tt} P_t\big)^\top (w_t)_j.$$

Write $\tilde{Z}_t = [\tilde{z}_{t,1}, \dots, \tilde{z}_{t,n_t}]$. For every output column i,

$$(M_t^{\mathrm{self},k+1})_{:,i} = \sum_{j=1}^{n_t} u_{t,j}^{k+1} \big((P_t)^\top H_{ij}^{tt} P_t\big) \xi_t^{k+1}.$$

Hence

$$w_{t,i}^{\mathrm{self},k+1} = \big((M_t^{\mathrm{self},k+1})_{:,i}\big)^\top(\tilde{Z}_t u_t^{k+1}) = \sum_{j=1}^{n_t} u_{t,j}^{k+1}(\xi_t^{k+1})^\top\big((P_t)^\top H_{ij}^{tt} P_t\big)^\top(\tilde{Z}_t u_t^{k+1}),$$

and therefore

$$\xi_t^{k+1} w_{t,i}^{\mathrm{self},k+1} = \sum_{j=1}^{n_t} u_{t,j}^{k+1}\big(\xi_t^{k+1}(\xi_t^{k+1})^\top\big)\big((P_t)^\top H_{ij}^{tt} P_t\big)^\top(\tilde{Z}_t u_t^{k+1}).$$

Taking $\mathbb{E}_{\xi_t}[\cdot \mid \mathcal{F}_k^+, u_t^{k+1}]$ and using $\mathbb{E}[\xi_t^{k+1}(\xi_t^{k+1})^\top] = I_{r_t}$ gives

$$\mathbb{E}_{\xi_t}\Big[\xi_t^{k+1} w_{t,i}^{\mathrm{self},k+1} \mid \mathcal{F}_k^+, u_t^{k+1}\Big] = \sum_{j=1}^{n_t} u_{t,j}^{k+1}\big((P_t)^\top H_{ij}^{tt} P_t\big)^\top(\tilde{Z}_t u_t^{k+1}).$$

Now take $\mathbb{E}_{u_t}[\cdot \mid \mathcal{F}_k^+]$ and use

$$\mathbb{E}\Big[u_{t,j}^{k+1}(\tilde{Z}_t u_t^{k+1}) \mid \mathcal{F}_k^+\Big] = \tilde{z}_{t,j}$$

to obtain

$$\mathbb{E}\Big[\xi_t^{k+1} w_{t,i}^{\mathrm{self},k+1} \mid \mathcal{F}_k^+\Big] = \sum_{j=1}^{n_t}\big((P_t)^\top H_{ij}^{tt} P_t\big)^\top \tilde{z}_{t,j} = (\bar{H}_{tt}^\sharp[\tilde{Z}_t])_{:,i}.$$

Therefore the entire rank-one matrix satisfies

$$\mathbb{E}[\widehat{B}_{\sharp,t}^{k+1} \mid \mathcal{F}_k^+] = P_t\, \bar{H}_{tt}^\sharp[\tilde{Z}_t]. \tag{38}$$

Applying Lemma 3(ii) to the matrices $(H_{ij}^{tt})^\top$ gives

$$\mathbb{E}[\widehat{B}_{\sharp,t}^{k+1} \mid \mathcal{F}_k] = b_t H_{tt}[Z_t] + (a_t - b_t) H_{tt}^\sharp[Z_t] + b_t Z_t T_t. \tag{39}$$

Step 3: self-layer linear inversion. Let

$$X_t := H_{tt}[Z_t], \quad X_t^\sharp := H_{tt}^\sharp[Z_t], \quad C_t := Z_t T_t.$$

By (35), (37), and (39),

$$\mathbb{E}[\bar{A}_t^{(t)} \mid \mathcal{F}_k] = (a_t - b_t)X_t + b_t X_t^\sharp + b_t C_t,$$
$$\mathbb{E}[\widehat{B}_{\sharp,t}^{k+1} \mid \mathcal{F}_k] = b_t X_t + (a_t - b_t)X_t^\sharp + b_t C_t, \quad \mathbb{E}[\widehat{C}_t^{k+1} \mid \mathcal{F}_k] = C_t.$$

Therefore the diagonal-block correction used in $\widehat{\mathcal{H}_{tt}[Z_t]}^{k+1}$ satisfies

$$\mathbb{E}[\widehat{\mathcal{H}_{tt}[Z_t]}^{k+1} \mid \mathcal{F}_k] = H_{tt}[Z_t]. \tag{40}$$

Step 4: pass from $\bar{\mathcal{H}}$ to the stochastic operator $\tilde{\mathcal{H}}^{k+1}$. The actual algorithm uses the shared random operator $\tilde{\mathcal{H}}^{k+1}$. All queried directions are $\mathcal{F}_k^{++}$-measurable:

$$E_t(\tilde{Z}_t) \in \mathcal{B} \quad \text{is } \mathcal{F}_k^+\text{-measurable}, \quad \Delta\boldsymbol{B}^{k+1} \in \mathcal{B} \quad \text{is } \mathcal{F}_k^{++}\text{-measurable}.$$

By Assumption 5,

$$\mathbb{E}[\tilde{\mathcal{H}}^{k+1}[W] \mid \mathcal{F}_k^{++}] = \bar{\mathcal{H}}[W] \quad \forall\, \mathcal{F}_k^{++}\text{-measurable } W \in \mathcal{B}.$$

Because $E_t(\tilde{Z}_t)$ is $\mathcal{F}_k^+$-measurable and therefore $\mathcal{F}_k^{++}$-measurable, we have

$$\mathbb{E}[\tilde{\mathcal{H}}^{k+1}[E_t(\tilde{Z}_t)] \mid \mathcal{F}_k^{++}] = \bar{\mathcal{H}}[E_t(\tilde{Z}_t)].$$

Multiplying by the deterministic lift $\boldsymbol{P}$ and then taking $\mathbb{E}[\cdot \mid \mathcal{F}_k]$ yields

$$\mathbb{E}[\boldsymbol{A}^{(t),k+1} \mid \mathcal{F}_k] = \mathbb{E}[\bar{\boldsymbol{A}}^{(t)} \mid \mathcal{F}_k].$$

Likewise, once the probes are fixed, the maps

$$M_t \longmapsto Z_t\big((M_t^\top \xi_t^{k+1})(u_t^{k+1})^\top\big), \quad M_t \longmapsto P_t\big(\xi_t^{k+1}((M_t^\top(\tilde{Z}_t u_t^{k+1}))^\top)\big)$$

are linear in the queried output block M_t. Since $\Delta \boldsymbol{B}^{k+1}$ is $\mathcal{F}_k^{++}$-measurable,

$$\mathbb{E}[\tilde{\mathcal{H}}^{k+1}[\Delta \boldsymbol{B}^{k+1}] \mid \mathcal{F}_k^{++}] = \bar{\mathcal{H}}[\Delta \boldsymbol{B}^{k+1}],$$

and therefore

$$\mathbb{E}[\widehat{C}_t^{k+1} \mid \mathcal{F}_k] = \mathbb{E}[\bar{\widehat{C}}_t^{k+1} \mid \mathcal{F}_k], \quad \mathbb{E}[\widehat{B}_{\sharp,t}^{k+1} \mid \mathcal{F}_k] = \mathbb{E}[\bar{\widehat{B}}_{\sharp,t}^{k+1} \mid \mathcal{F}_k],$$

where the barred quantities are computed with the deterministic operator $\bar{\mathcal{H}}$. Consequently the deterministic first-moment identities derived in Steps 1–3 pass unchanged to the stochastic estimator built from $\tilde{\mathcal{H}}^{k+1}$. In particular,

$$\mathbb{E}[(A_\ell^{(t),k+1})_{\ell \neq t} \mid \mathcal{F}_k] = H_{\ell t}[Z_t], \quad \mathbb{E}[\widehat{\mathcal{H}_{tt}[Z_t]}^{k+1} \mid \mathcal{F}_k] = H_{tt}[Z_t].$$

Summing the corrected diagonal and lifted off-diagonal block contributions yields

$$\mathbb{E}[\widehat{\mathcal{H}[\boldsymbol{Z}]}^{k+1} \mid \mathcal{F}_k] = H[\boldsymbol{Z}]. \tag{41}$$

Finally, using Assumption 5 and Lemma 3(iv),

$$\mathbb{E}[\boldsymbol{P}\tilde{\boldsymbol{U}}_Y^{k+1} \mid \mathcal{F}_k] = \nabla_{\boldsymbol{Y}} f(x^k, \boldsymbol{Y}),$$

so

$$\mathbb{E}[\widehat{\boldsymbol{S}}^{k+1} \mid \mathcal{F}_k] = H[\boldsymbol{Z}] - f_Y.$$

Part (ii): second moments. We first prove (34a) and (34b), and then derive (34c).

Step 1: bound for $\widehat{\boldsymbol{V}}^{k+1}$. Let $\boldsymbol{G} := \nabla_{\boldsymbol{Y}} g(x^k, \boldsymbol{Y})$. Define the block-diagonal lifted projector

$$(\boldsymbol{Q}\boldsymbol{U})_\ell := Q_\ell U_\ell, \quad \boldsymbol{U} \in \mathcal{Y}.$$

Then

$$\widehat{\boldsymbol{V}}^{k+1} - \boldsymbol{G} = \boldsymbol{P}\big(\tilde{\boldsymbol{V}}^{k+1} - \boldsymbol{P}^\top \boldsymbol{G}\big) + (\boldsymbol{Q} - I)\boldsymbol{G}.$$

Conditioned on $\mathcal{F}_k^+$, the first term has mean zero. Using

$$\|\boldsymbol{P}\boldsymbol{U}\|^2 = \sum_{\ell=1}^L \|P_\ell U_\ell\|^2 \le \omega_P \sum_{\ell=1}^L \|U_\ell\|^2 = \omega_P \|\boldsymbol{U}\|^2,$$

Assumption 5 gives

$$\mathbb{E}\left[\left\|\boldsymbol{P}\big(\tilde{\boldsymbol{V}}^{k+1} - \boldsymbol{P}^\top \boldsymbol{G}\big)\right\|^2 \mid \mathcal{F}_k^+\right] \le \omega_P \tilde{\sigma}_{g,1}^2.$$

Also,

$$\mathbb{E}[\|(\boldsymbol{Q} - I)\boldsymbol{G}\|^2 \mid \mathcal{F}_k] = \sum_{\ell=1}^L \mathbb{E}[\|(Q_\ell - I)G_\ell\|^2 \mid \mathcal{F}_k] \le (\omega_P - 1)\|\boldsymbol{G}\|^2.$$

Since $\nabla_{\boldsymbol{Y}} g(x^k, \boldsymbol{Y}^*(x^k)) = 0$ and ∇g is $L_{g,1}$-Lipschitz,

$$\|\boldsymbol{G}\| \le L_{g,1}\|\boldsymbol{Y} - \boldsymbol{Y}^*(x^k)\|.$$

This proves (34a).

Step 2: bound for $\widehat{W}^{k+1}$. Write

$$R_x := \tilde{U}_x^{k+1}, \quad R_J := \tilde{\mathcal{J}}^{k+1}[\boldsymbol{P}^\top \boldsymbol{Z}].$$

By the law of total variance,

$$\begin{aligned}\mathbb{E}[\|R_J - \mathbb{E}[R_J \mid \mathcal{F}_k]\|^2 \mid \mathcal{F}_k] &= \mathbb{E}[\|R_J - \mathbb{E}[R_J \mid \mathcal{F}_k^+]\|^2 \mid \mathcal{F}_k] \\ &\quad + \mathbb{E}[\|\mathbb{E}[R_J \mid \mathcal{F}_k^+] - \mathbb{E}[R_J \mid \mathcal{F}_k]\|^2 \mid \mathcal{F}_k].\end{aligned}$$

For the first term, Assumption 5 gives

$$\mathbb{E}[\|R_J - \mathbb{E}[R_J \mid \mathcal{F}_k^+]\|^2 \mid \mathcal{F}_k^+] \le \tilde{\sigma}_{g,2}^2 \|\boldsymbol{P}^\top \boldsymbol{Z}\|^2.$$

Taking expectation over $\mathcal{F}_k^+$ and using $\mathbb{E}[\|\boldsymbol{P}^\top \boldsymbol{Z}\|^2 \mid \mathcal{F}_k] = \|\boldsymbol{Z}\|^2$ yields

$$\mathbb{E}[\|R_J - \mathbb{E}[R_J \mid \mathcal{F}_k^+]\|^2 \mid \mathcal{F}_k] \le \tilde{\sigma}_{g,2}^2 \|\boldsymbol{Z}\|^2.$$

For the second term,

$$\mathbb{E}[R_J \mid \mathcal{F}_k^+] = \mathcal{J}(x^k, \boldsymbol{Y})[\boldsymbol{Q}\boldsymbol{Z}], \quad \mathbb{E}[R_J \mid \mathcal{F}_k] = \mathcal{J}(x^k, \boldsymbol{Y})[\boldsymbol{Z}].$$

Hence

$$\begin{aligned}\mathbb{E}[\|\mathbb{E}[R_J \mid \mathcal{F}_k^+] - \mathbb{E}[R_J \mid \mathcal{F}_k]\|^2 \mid \mathcal{F}_k] &= \mathbb{E}[\|\mathcal{J}(x^k, \boldsymbol{Y})[(\boldsymbol{Q} - I)\boldsymbol{Z}]\|^2 \mid \mathcal{F}_k] \\ &\le L_{g,1}^2 \mathbb{E}[\|(\boldsymbol{Q} - I)\boldsymbol{Z}\|^2 \mid \mathcal{F}_k] \le \Lambda_Y \|\boldsymbol{Z}\|^2.\end{aligned}$$

The oracle $\tilde{U}_x^{k+1}$ is independent of $\tilde{\mathcal{J}}^{k+1}$ given $\mathcal{F}_k^+$, so

$$\mathbb{E}[\|R_x - \mathbb{E}[R_x \mid \mathcal{F}_k]\|^2 \mid \mathcal{F}_k] \le \tilde{\sigma}_{f,1}^2.$$

Combining the three displays yields (34b).

Step 3: bound for $\widehat{\boldsymbol{S}}^{k+1}$. Define

$$\Delta_H^{k+1} := \widehat{\mathcal{H}[\boldsymbol{Z}]}^{k+1} - H[\boldsymbol{Z}], \quad \Delta_U^{k+1} := \boldsymbol{P}\tilde{\boldsymbol{U}}_Y^{k+1} - f_Y.$$

Then

$$\Delta_S^{k+1} = \Delta_H^{k+1} - \Delta_U^{k+1},$$

so

$$\|\Delta_S^{k+1}\|^2 \le 2\|\Delta_H^{k+1}\|^2 + 2\|\Delta_U^{k+1}\|^2. \tag{42}$$

Step 3a: bound for Δ_U^{k+1}. Exactly as in Step 1,

$$\Delta_U^{k+1} = \boldsymbol{P}(\tilde{\boldsymbol{U}}_Y^{k+1} - \boldsymbol{P}^\top f_Y) + (\boldsymbol{Q} - I) f_Y.$$

Conditioned on $\mathcal{F}_k^+$, the first term has mean zero, and therefore

$$\mathbb{E}[\|\boldsymbol{P}(\tilde{\boldsymbol{U}}_Y^{k+1} - \boldsymbol{P}^\top f_Y)\|^2 \mid \mathcal{F}_k] \le \omega_P \tilde{\sigma}_{f,2}^2.$$

Also,

$$\mathbb{E}[\|(\boldsymbol{Q} - I) f_Y\|^2 \mid \mathcal{F}_k] \le (\omega_P - 1)\|f_Y\|^2.$$

Since

$$\|f_Y\| \le \|\nabla_{\boldsymbol{Y}} f(x^k, \boldsymbol{Y}^*(x^k))\| + L_{f,1}\|\boldsymbol{Y} - \boldsymbol{Y}^*(x^k)\| \le L_{f,0} + L_{f,1}\|\boldsymbol{Y} - \boldsymbol{Y}^*(x^k)\|,$$

we obtain

$$\|f_Y\|^2 \le 2L_{f,0}^2 + 2L_{f,1}^2 \|\boldsymbol{Y} - \boldsymbol{Y}^*(x^k)\|^2.$$

Consequently,

$$\mathbb{E}[\|\Delta_U^{k+1}\|^2 \mid \mathcal{F}_k] \le \omega_P \tilde{\sigma}_{f,2}^2 + 2(\omega_P - 1)L_{f,0}^2 + 2(\omega_P - 1)L_{f,1}^2 \|\boldsymbol{Y} - \boldsymbol{Y}^*(x^k)\|^2. \tag{43}$$

Step 3b: bound for Δ_H^{k+1}. For each layer t, define the centered contribution $\delta_t^{k+1} \in \mathcal{Y}$ by

$$(\delta_t^{k+1})_\ell := \begin{cases} A_\ell^{(t),k+1} - H_{\ell t}[Z_t], & \ell \ne t, \\ \widehat{\mathcal{H}_{tt}[Z_t]}^{k+1} - H_{tt}[Z_t], & \ell = t. \end{cases}$$

Then $\Delta_H^{k+1} = \sum_{t=1}^L \delta_t^{k+1}$. Hence

$$\|\Delta_H^{k+1}\|^2 \le L \sum_{t=1}^L \|\delta_t^{k+1}\|^2. \tag{44}$$

For each layer t, define the self-layer correction coefficients

$$c_{A,t} := \frac{a_t - b_t}{a_t(a_t - 2b_t)}, \quad c_{B,t} := \frac{b_t}{a_t(a_t - 2b_t)}, \quad c_{C,t} := \frac{b_t}{a_t},$$

and let

$$\bar{c}_A := \max_{1\le t\le L} c_{A,t}, \quad \bar{c}_B := \max_{1\le t\le L} c_{B,t}, \quad \bar{c}_C := \max_{1\le t\le L} c_{C,t}.$$

Because $\mathbb{E}[\widehat{\mathcal{H}_{tt}[Z_t]}^{k+1} \mid \mathcal{F}_k] = H_{tt}[Z_t]$, we may write

$$\begin{aligned}\widehat{\mathcal{H}_{tt}[Z_t]}^{k+1} - H_{tt}[Z_t] = c_{A,t}\big(A_t^{(t),k+1} - \mathbb{E}[A_t^{(t),k+1} \mid \mathcal{F}_k]\big) - c_{B,t}\big(\widehat{B}_{\sharp,t}^{k+1} - \mathbb{E}[\widehat{B}_{\sharp,t}^{k+1} \mid \mathcal{F}_k]\big) \\ - c_{C,t}\big(\widehat{C}_t^{k+1} - \mathbb{E}[\widehat{C}_t^{k+1} \mid \mathcal{F}_k]\big).\end{aligned}$$

Using $\mathbb{E}[\|X - \mathbb{E}[X]\|^2] \le \mathbb{E}[\|X\|^2]$ and $\|a+b+c\|^2 \le 3(\|a\|^2 + \|b\|^2 + \|c\|^2)$, we obtain

$$\mathbb{E}[\|\delta_t^{k+1}\|^2 \mid \mathcal{F}_k] \le (1 + 3\bar{c}_A^2)\mathbb{E}[\|\boldsymbol{A}^{(t),k+1}\|^2 \mid \mathcal{F}_k] + 3\bar{c}_B^2\mathbb{E}[\|\widehat{B}_{\sharp,t}^{k+1}\|^2 \mid \mathcal{F}_k] + 3\bar{c}_C^2\mathbb{E}[\|\widehat{C}_t^{k+1}\|^2 \mid \mathcal{F}_k].$$

We now bound the three second moments. First, conditioned on $\mathcal{F}_k^+$,

$$\|\bar{\mathcal{H}}[W]\| \le \omega_P L_{g,1}\|W\|, \quad \forall W \in \mathcal{B},$$

because $\|\boldsymbol{P}\| \le \sqrt{\omega_P}$ and $\|\boldsymbol{P}^\top\| \le \sqrt{\omega_P}$. Using Assumption 5 and $\|\boldsymbol{P}U\|^2 \le \omega_P\|U\|^2$,

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{A}^{(t),k+1}\|^2 \mid \mathcal{F}_k^+] &= \mathbb{E}[\|\boldsymbol{P}\tilde{\mathcal{H}}^{k+1}[E_t(\tilde{Z}_t)]\|^2 \mid \mathcal{F}_k^+] \\ &\le \omega_P\mathbb{E}[\|\tilde{\mathcal{H}}^{k+1}[E_t(\tilde{Z}_t)]\|^2 \mid \mathcal{F}_k^+] \\ &\le 2\omega_P\big(\tilde{\sigma}_{g,2}^2 + \omega_P^2 L_{g,1}^2\big)\|\tilde{Z}_t\|^2.\end{aligned}$$

Taking expectation over $\mathcal{F}_k^+$ and using $\mathbb{E}[\|\tilde{Z}_t\|^2 \mid \mathcal{F}_k] = \|Z_t\|^2$ yields

$$\mathbb{E}[\|\boldsymbol{A}^{(t),k+1}\|^2 \mid \mathcal{F}_k] \le \Lambda_A\|Z_t\|^2, \quad \Lambda_A := 2\omega_P\big(\tilde{\sigma}_{g,2}^2 + \omega_P^2 L_{g,1}^2\big).$$

Next define $D_{rn} := \sum_{\ell=1}^L r_\ell n_\ell$ and $\eta_{rn} := \max_{1\le\ell\le L} r_\ell n_\ell$. Since each probe block is rank one with Rademacher entries,

$$\|\Delta\boldsymbol{B}^{k+1}\|^2 = \sum_{\ell=1}^L \|\xi_\ell^{k+1}(u_\ell^{k+1})^\top\|_F^2 = D_{rn}.$$

Therefore, conditioned on $\mathcal{F}_k^+$,

$$\mathbb{E}[\|\boldsymbol{M}^{k+1}\|^2 \mid \mathcal{F}_k^+] \le 2\big(\tilde{\sigma}_{g,2}^2 + \omega_P^2 L_{g,1}^2\big)D_{rn},$$

and hence, for each layer t,

$$\mathbb{E}[\|M_t^{k+1}\|^2 \mid \mathcal{F}_k^+] \le 2\big(\tilde{\sigma}_{g,2}^2 + \omega_P^2 L_{g,1}^2\big)D_{rn}.$$

For $\widehat{C}_t^{k+1}$, since

$$\widehat{C}_t^{k+1} = Z_t\big(v_t^{k+1}(u_t^{k+1})^\top\big)$$

and $\|u_t^{k+1}\|^2 = n_t$, we have

$$\|\widehat{C}_t^{k+1}\|^2 = \|Z_t v_t^{k+1}\|^2\|u_t^{k+1}\|^2 \le n_t\|Z_t\|^2\|v_t^{k+1}\|^2.$$

Moreover,

$$v_t^{k+1} = (M_t^{k+1})^\top\xi_t^{k+1}, \quad \|v_t^{k+1}\|^2 \le r_t\|M_t^{k+1}\|^2.$$

Hence

$$\mathbb{E}[\|\widehat{C}_t^{k+1}\|^2 \mid \mathcal{F}_k] \le \Lambda_C\|Z_t\|^2, \quad \Lambda_C := 2\eta_{rn}D_{rn}\big(\tilde{\sigma}_{g,2}^2 + \omega_P^2 L_{g,1}^2\big).$$

For $\widehat{B}_{\sharp,t}^{k+1}$,

$$\widehat{B}_{\sharp,t}^{k+1} = P_t\big(\xi_t^{k+1}(w_t^{k+1})^\top\big),$$

so

$$\|\widehat{B}_{\sharp,t}^{k+1}\|^2 \le \omega_P \|\xi_t^{k+1}(w_t^{k+1})^\top\|^2 = \omega_P r_t \|w_t^{k+1}\|^2.$$

Since

$$w_t^{k+1} = (M_t^{k+1})^\top(\tilde{Z}_t u_t^{k+1}),$$

we get

$$\|w_t^{k+1}\|^2 \le \|M_t^{k+1}\|^2 \|\tilde{Z}_t u_t^{k+1}\|^2 \le n_t \|M_t^{k+1}\|^2 \|\tilde{Z}_t\|^2.$$

Conditioning on $\mathcal{F}_k^+$ and using the bound for $\mathbb{E}[\|M_t^{k+1}\|^2 \mid \mathcal{F}_k^+]$ gives

$$\mathbb{E}[\|\widehat{B}_{\sharp,t}^{k+1}\|^2 \mid \mathcal{F}_k^+] \le 2\omega_P \eta_{rn} D_{rn}\big(\tilde{\sigma}_{g,2}^2 + \omega_P^2 L_{g,1}^2\big)\|\tilde{Z}_t\|^2.$$

Taking expectation over $\mathcal{F}_k^+$ and using $\mathbb{E}[\|\tilde{Z}_t\|^2 \mid \mathcal{F}_k] = \|Z_t\|^2$ yields

$$\mathbb{E}[\|\widehat{B}_{\sharp,t}^{k+1}\|^2 \mid \mathcal{F}_k] \le \Lambda_B \|Z_t\|^2,$$

where $\Lambda_B := 2\omega_P \eta_{rn} D_{rn}\big(\tilde{\sigma}_{g,2}^2 + \omega_P^2 L_{g,1}^2\big)$.
Combining the last three displays with (44) shows

$$\mathbb{E}[\|\Delta_H^{k+1}\|^2 \mid \mathcal{F}_k] \le \Lambda_H \|\boldsymbol{Z}\|^2,$$

where one may take

$$\Lambda_H := L\Big((1 + 3\bar{c}_A^2)\Lambda_A + 3\bar{c}_B^2\Lambda_B + 3\bar{c}_C^2\Lambda_C\Big). \tag{45}$$

Finally, combining (42), (43), and the bound on Δ_H^{k+1} gives

$$\mathbb{E}[\|\Delta_S^{k+1}\|^2 \mid \mathcal{F}_k] \le 2\Lambda_H \|\boldsymbol{Z}\|^2 + 2\omega_P \tilde{\sigma}_{f,2}^2 + 4(\omega_P - 1)L_{f,0}^2 + 4(\omega_P - 1)L_{f,1}^2 \|\boldsymbol{Y} - \boldsymbol{Y}^*(x^k)\|^2.$$

Therefore (34c) holds with

$$\sigma_{S,0}^2 := 2\omega_P \tilde{\sigma}_{f,2}^2 + 4(\omega_P - 1)L_{f,0}^2, \quad \sigma_{S,2}^2 := 4(\omega_P - 1)L_{f,1}^2, \quad \Lambda_S := 2\Lambda_H.$$

□

Lemma 6 (Tracking bounds for $\boldsymbol{Y}^k$ and $\boldsymbol{Z}^k$)**.** *Assume* (A1)–(A3) *and the sampling conditions of Lemma 5. Let*

$$\boldsymbol{Y}_*^k := \boldsymbol{Y}^*(x^k), \quad \boldsymbol{Z}_*^k := \boldsymbol{Z}^*(x^k).$$

Define $x_+^k := x^k - h^k$ *so that*

$$x^{k+1} - x^k = \alpha_k(x_+^k - x^k), \tag{46}$$

and couple the stepsizes as

$$\beta_k = c_1\alpha_k, \quad \gamma_k = c_2\alpha_k \quad (\forall k \ge 0). \tag{47}$$

Assume further that

$$\beta_k \le \bar{\beta}, \tag{48}$$

$$\gamma_k \le \bar{\gamma}, \tag{49}$$

for sufficiently small constants $\bar{\beta}, \bar{\gamma}$ *depending only on the constants in Lemmas 2 and 5. Then there exist finite constants* $C_{Yx}, C_{Y,0}, C_{Y,1}, C_{Zx}, C_{Z,0}, C_{Z,1}$ *such that for all* $K \ge 0$*,*

$$\sum_{k=0}^K \alpha_k \,\mathbb{E}\|\boldsymbol{Y}^k - \boldsymbol{Y}_*^k\|^2 \le C_{Yx} \sum_{k=0}^K \alpha_k \,\mathbb{E}\|x_+^k - x^k\|^2 + C_{Y,0} + C_{Y,1} \sum_{k=0}^K \alpha_k^2, \tag{50}$$

$$\sum_{k=0}^K \alpha_k \,\mathbb{E}\|\boldsymbol{Z}^k - \boldsymbol{Z}_*^k\|^2 \le C_{Zx} \sum_{k=0}^K \alpha_k \,\mathbb{E}\|x_+^k - x^k\|^2 + C_{Z,0} + C_{Z,1} \sum_{k=0}^K \alpha_k^2. \tag{51}$$

Proof. Define the tracking errors

$$E_Y^k := \boldsymbol{Y}^k - \boldsymbol{Y}_*^k, \quad E_Z^k := \boldsymbol{Z}^k - \boldsymbol{Z}_*^k.$$

We also introduce the constants

$$\sigma_Y^2 := \omega_P \tilde{\sigma}_{g,1}^2, \quad \Lambda_Y := (\omega_P - 1) L_{g,1}^2,$$

$$B_{ZY} := L_{f,1} + \frac{L_{g,2} L_{f,0}}{\mu_g}, \quad \Lambda_Z := 2\Lambda_S, \quad \bar{\sigma}_{S,0}^2 := \sigma_{S,0}^2 + \frac{2\Lambda_S L_{f,0}^2}{\mu_g^2},$$

where $\sigma_{S,0}, \sigma_{S,2}, \Lambda_S$ are any valid constants from Lemma 5. We repeatedly condition on $\mathcal{F}_k$ and work in the product Frobenius space $\mathcal{Y}$.

Part I: tracking of $\boldsymbol{Y}^k$. The lower update is $\boldsymbol{Y}^{k+1} = \boldsymbol{Y}^k - \beta_k \widehat{\boldsymbol{V}}^{k+1}$ and Lemma 5 gives

$$\mathbb{E}[\widehat{\boldsymbol{V}}^{k+1} \mid \mathcal{F}_k] = \nabla_{\boldsymbol{Y}} g(x^k, \boldsymbol{Y}^k).$$

Step 1: drift of the target point $\boldsymbol{Y}_*^k$. Using $\|U+V\|^2 \le (1+c)\|U\|^2 + (1+1/c)\|V\|^2$ with

$$U := \boldsymbol{Y}^{k+1} - \boldsymbol{Y}_*^k, \quad V := \boldsymbol{Y}_*^k - \boldsymbol{Y}_*^{k+1}, \quad c := \beta_k \mu_g,$$

we obtain

$$\|E_Y^{k+1}\|^2 \le (1+\beta_k\mu_g)\|\boldsymbol{Y}^{k+1} - \boldsymbol{Y}_*^k\|^2 + \left(1 + \frac{1}{\beta_k\mu_g}\right)\|\boldsymbol{Y}_*^{k+1} - \boldsymbol{Y}_*^k\|^2. \tag{52}$$

By (16) and (46),

$$\|\boldsymbol{Y}_*^{k+1} - \boldsymbol{Y}_*^k\| \le L_{Y^*}\|x^{k+1} - x^k\| = \alpha_k L_{Y^*}\|x_+^k - x^k\|.$$

Substituting into (52) yields

$$\|E_Y^{k+1}\|^2 \le (1+\beta_k\mu_g)\|\boldsymbol{Y}^{k+1} - \boldsymbol{Y}_*^k\|^2 + \left(\alpha_k^2 + \frac{\alpha_k^2}{\beta_k\mu_g}\right) L_{Y^*}^2 \|x_+^k - x^k\|^2. \tag{53}$$

Step 2: recursion for $\|\boldsymbol{Y}^{k+1} - \boldsymbol{Y}_*^k\|^2$. Define

$$\Xi_V^{k+1} := \widehat{\boldsymbol{V}}^{k+1} - \nabla_{\boldsymbol{Y}} g(x^k, \boldsymbol{Y}^k), \quad \mathbb{E}[\Xi_V^{k+1} \mid \mathcal{F}_k] = 0.$$

Then

$$\mathbb{E}[\|\boldsymbol{Y}^{k+1} - \boldsymbol{Y}_*^k\|^2 \mid \mathcal{F}_k] = \|\boldsymbol{Y}^k - \boldsymbol{Y}_*^k - \beta_k \nabla_{\boldsymbol{Y}} g(x^k, \boldsymbol{Y}^k)\|^2 + \beta_k^2 \mathbb{E}[\|\Xi_V^{k+1}\|^2 \mid \mathcal{F}_k]. \tag{54}$$

Because $\boldsymbol{Y} \mapsto g(x^k, \boldsymbol{Y})$ is μ_g-strongly convex with $L_{g,1}$-Lipschitz gradient on $\mathcal{Y}$, the deterministic gradient step contracts whenever $\beta_k \le \frac{2}{\mu_g + L_{g,1}}$:

$$\|\boldsymbol{Y}^k - \boldsymbol{Y}_*^k - \beta_k \nabla_{\boldsymbol{Y}} g(x^k, \boldsymbol{Y}^k)\|^2 \le (1 - \beta_k\mu_g)^2 \|E_Y^k\|^2. \tag{55}$$

Also, Lemma 5 gives

$$\mathbb{E}[\|\Xi_V^{k+1}\|^2 \mid \mathcal{F}_k] \le \sigma_Y^2 + \Lambda_Y \|E_Y^k\|^2. \tag{56}$$

Combining (54)–(56) yields

$$\mathbb{E}[\|\boldsymbol{Y}^{k+1} - \boldsymbol{Y}_*^k\|^2 \mid \mathcal{F}_k] \le \big((1-\beta_k\mu_g)^2 + \beta_k^2\Lambda_Y\big)\|E_Y^k\|^2 + \beta_k^2\sigma_Y^2. \tag{57}$$

Step 3: combine with the drift bound. Choose $\bar{\beta}$ small enough that $\bar{\beta} \le \min\left\{\frac{2}{\mu_g + L_{g,1}}, \frac{1}{4\mu_g}, \frac{\mu_g}{8\Lambda_Y}\right\}$ with the convention $\mu_g/(8\Lambda_Y) = +\infty$ when $\Lambda_Y = 0$. Then for every $\beta_k \le \bar{\beta}$,

$$(1+\beta_k\mu_g)\big((1-\beta_k\mu_g)^2 + \beta_k^2\Lambda_Y\big) \le 1 - \frac{\beta_k\mu_g}{4},$$

$$1 + \beta_k\mu_g \le 2, \quad \alpha_k^2 + \frac{\alpha_k^2}{\beta_k\mu_g} \le \frac{2\alpha_k^2}{\beta_k\mu_g}.$$

Taking $\mathbb{E}[\cdot \mid \mathcal{F}_k]$ in (53) and using (57) gives

$$\mathbb{E}[\|E_Y^{k+1}\|^2 \mid \mathcal{F}_k] \le \left(1 - \frac{\beta_k \mu_g}{4}\right) \|E_Y^k\|^2 + \frac{2\alpha_k^2 L_{Y^*}^2}{\beta_k \mu_g} \|x_+^k - x^k\|^2 + 2\beta_k^2 \sigma_Y^2. \tag{58}$$

Rearranging, taking total expectation, and summing over $k = 0, \dots, K$ yield

$$\frac{\mu_g}{4} \sum_{k=0}^{K} \beta_k \, \mathbb{E}\|E_Y^k\|^2 \le \mathbb{E}\|E_Y^0\|^2 + \frac{2L_{Y^*}^2}{\mu_g} \sum_{k=0}^{K} \frac{\alpha_k^2}{\beta_k} \mathbb{E}\|x_+^k - x^k\|^2 + 2\sigma_Y^2 \sum_{k=0}^{K} \beta_k^2. \tag{59}$$

Using $\beta_k = c_1 \alpha_k$ gives

$$\sum_{k=0}^{K} \alpha_k \, \mathbb{E}\|E_Y^k\|^2 \le \frac{8L_{Y^*}^2}{c_1^2 \mu_g^2} \sum_{k=0}^{K} \alpha_k \, \mathbb{E}\|x_+^k - x^k\|^2 + \frac{4}{c_1 \mu_g} \mathbb{E}\|E_Y^0\|^2 + \frac{8c_1}{\mu_g} \sigma_Y^2 \sum_{k=0}^{K} \alpha_k^2.$$

This proves (50); in particular one may take

$$C_{Yx} := \frac{8L_{Y^*}^2}{c_1^2 \mu_g^2}, \quad C_{Y,0} := \frac{4}{c_1 \mu_g} \mathbb{E}\|E_Y^0\|^2, \quad C_{Y,1} := \frac{8c_1}{\mu_g} \sigma_Y^2.$$

Part II: tracking of $\boldsymbol{Z}^k$. The auxiliary update is $\boldsymbol{Z}^{k+1} = \boldsymbol{Z}^k - \gamma_k \widehat{\boldsymbol{S}}^{k+1}$ and Lemma 5 gives

$$\mathbb{E}[\widehat{\boldsymbol{S}}^{k+1} \mid \mathcal{F}_k] = \mathcal{H}(x^k, \boldsymbol{Y}^k)[\boldsymbol{Z}^k] - \nabla_{\boldsymbol{Y}} f(x^k, \boldsymbol{Y}^k).$$

Write

$$H^k := \mathcal{H}(x^k, \boldsymbol{Y}^k), \quad f_Y^k := \nabla_{\boldsymbol{Y}} f(x^k, \boldsymbol{Y}^k).$$

Step 1: drift of the target point $\boldsymbol{Z}_*^k$. Using $\|U+V\|^2 \le (1+c)\|U\|^2 + (1+1/c)\|V\|^2$ with

$$U := \boldsymbol{Z}^{k+1} - \boldsymbol{Z}_*^k, \quad V := \boldsymbol{Z}_*^k - \boldsymbol{Z}_*^{k+1}, \quad c := \frac{\gamma_k \mu_g}{3},$$

we obtain

$$\|E_Z^{k+1}\|^2 \le \left(1 + \frac{\gamma_k \mu_g}{3}\right) \|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}_*^k\|^2 + \left(1 + \frac{3}{\gamma_k \mu_g}\right) \|\boldsymbol{Z}_*^{k+1} - \boldsymbol{Z}_*^k\|^2. \tag{60}$$

By (17) and (46),

$$\|\boldsymbol{Z}_*^{k+1} - \boldsymbol{Z}_*^k\| \le L_{Z^*} \|x^{k+1} - x^k\| = \alpha_k L_{Z^*} \|x_+^k - x^k\|.$$

Thus, provided $\gamma_k \mu_g \le 1/4$,

$$\mathbb{E}[\|E_Z^{k+1}\|^2 \mid \mathcal{F}_k] \le \left(1 + \frac{\gamma_k \mu_g}{3}\right) \mathbb{E}[\|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}_*^k\|^2 \mid \mathcal{F}_k] + \frac{4\alpha_k^2 L_{Z^*}^2}{\gamma_k \mu_g} \|x_+^k - x^k\|^2. \tag{61}$$

Step 2: recursion for $\|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}_*^k\|^2$. Let

$$\Xi_S^{k+1} := \widehat{\boldsymbol{S}}^{k+1} - (H^k[\boldsymbol{Z}^k] - f_Y^k), \quad \mathbb{E}[\Xi_S^{k+1} \mid \mathcal{F}_k] = 0.$$

Then

$$\mathbb{E}[\|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}_*^k\|^2 \mid \mathcal{F}_k] = \|\boldsymbol{Z}^k - \boldsymbol{Z}_*^k - \gamma_k (H^k[\boldsymbol{Z}^k] - f_Y^k)\|^2 + \gamma_k^2 \mathbb{E}[\|\Xi_S^{k+1}\|^2 \mid \mathcal{F}_k]. \tag{62}$$

Write $H_*^k := \mathcal{H}(x^k, \boldsymbol{Y}_*^k)$ and $f_{Y,*}^k := \nabla_{\boldsymbol{Y}} f(x^k, \boldsymbol{Y}_*^k)$, so that

$$H_*^k[\boldsymbol{Z}_*^k] - f_{Y,*}^k = 0.$$

Define $B_Z^k := (H^k - H_*^k)[\boldsymbol{Z}_*^k] - (f_Y^k - f_{Y,*}^k)$. Then

$$H^k[\boldsymbol{Z}^k] - f_Y^k = H^k[E_Z^k] + B_Z^k,$$

hence

$$\boldsymbol{Z}^k - \boldsymbol{Z}_*^k - \gamma_k (H^k[\boldsymbol{Z}^k] - f_Y^k) = (I - \gamma_k H^k)[E_Z^k] - \gamma_k B_Z^k.$$

Using $\|U+V\|^2 \le (1+c)\|U\|^2 + (1+1/c)\|V\|^2$ with $c = \gamma_k\mu_g/2$,

$$\|(I-\gamma_k H^k)[E_Z^k] - \gamma_k B_Z^k\|^2 \le \left(1+\frac{\gamma_k\mu_g}{2}\right)\|(I-\gamma_k H^k)[E_Z^k]\|^2 + \left(1+\frac{2}{\gamma_k\mu_g}\right)\gamma_k^2\|B_Z^k\|^2. \quad (63)$$

Because H^k is self-adjoint on $\mathcal{Y}$, $\mu_g I \preceq H^k \preceq L_{g,1} I$, and $\gamma_k \le 1/L_{g,1}$, we have

$$\|(I-\gamma_k H^k)[E_Z^k]\|^2 \le (1-\gamma_k\mu_g)^2\|E_Z^k\|^2. \quad (64)$$

Moreover, by Lipschitz continuity of $\nabla^2 g$ and ∇f and the bound (20),

$$\|B_Z^k\| \le \left(L_{g,2}\|\boldsymbol{Z}_*^k\| + L_{f,1}\right)\|E_Y^k\| \le B_{ZY}\|E_Y^k\|. \quad (65)$$

Lemma 5 gives

$$\mathbb{E}[\|\Xi_S^{k+1}\|^2 \mid \mathcal{F}_k] \le \sigma_{S,0}^2 + \sigma_{S,2}^2\|E_Y^k\|^2 + \Lambda_S\|\boldsymbol{Z}^k\|^2. \quad (66)$$

Since

$$\|\boldsymbol{Z}^k\|^2 \le 2\|E_Z^k\|^2 + 2\|\boldsymbol{Z}_*^k\|^2 \le 2\|E_Z^k\|^2 + \frac{2L_{f,0}^2}{\mu_g^2},$$

we deduce

$$\mathbb{E}[\|\Xi_S^{k+1}\|^2 \mid \mathcal{F}_k] \le \Lambda_Z\|E_Z^k\|^2 + \sigma_{S,2}^2\|E_Y^k\|^2 + \bar{\sigma}_{S,0}^2. \quad (67)$$

Step 3: collect terms. Substituting (63)–(67) into (62) yields

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}_*^k\|^2 \mid \mathcal{F}_k] \le{}& \left(\left(1+\frac{\gamma_k\mu_g}{2}\right)(1-\gamma_k\mu_g)^2 + \gamma_k^2\Lambda_Z\right)\|E_Z^k\|^2 \\ &+ \left(\left(1+\frac{2}{\gamma_k\mu_g}\right)\gamma_k^2 B_{ZY}^2 + \gamma_k^2\sigma_{S,2}^2\right)\|E_Y^k\|^2 + \gamma_k^2\bar{\sigma}_{S,0}^2.\end{aligned} \quad (68)$$

Choose $\bar{\gamma}$ small enough that $\bar{\gamma} \le \min\left\{\frac{1}{4\mu_g}, \frac{1}{L_{g,1}}, \frac{\mu_g}{16\Lambda_Z}\right\}$, with the convention $\mu_g/(16\Lambda_Z) = +\infty$ when $\Lambda_Z = 0$. Then for every $\gamma_k \le \bar{\gamma}$,

$$\left(1+\frac{\gamma_k\mu_g}{2}\right)(1-\gamma_k\mu_g)^2 + \gamma_k^2\Lambda_Z \le 1 - \frac{7\gamma_k\mu_g}{8},$$

and

$$\left(1+\frac{2}{\gamma_k\mu_g}\right)\gamma_k^2 \le \frac{9\gamma_k}{4\mu_g}, \quad \gamma_k^2\sigma_{S,2}^2 \le \frac{\gamma_k}{4\mu_g}\sigma_{S,2}^2.$$

Therefore

$$\mathbb{E}[\|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}_*^k\|^2 \mid \mathcal{F}_k] \le \left(1-\frac{7\gamma_k\mu_g}{8}\right)\|E_Z^k\|^2 + \frac{9B_{ZY}^2 + \sigma_{S,2}^2}{4\mu_g}\gamma_k\|E_Y^k\|^2 + \gamma_k^2\bar{\sigma}_{S,0}^2. \quad (69)$$

Step 4: combine with the drift bound. Combining (61) and (69), and using $\gamma_k\mu_g \le 1/4$, gives

$$\mathbb{E}[\|E_Z^{k+1}\|^2 \mid \mathcal{F}_k] \le \left(1-\frac{\gamma_k\mu_g}{4}\right)\|E_Z^k\|^2 + \frac{9B_{ZY}^2+\sigma_{S,2}^2}{2\mu_g}\gamma_k\|E_Y^k\|^2 + \frac{4\alpha_k^2 L_{Z^*}^2}{\gamma_k\mu_g}\|x_+^k - x^k\|^2 + 2\gamma_k^2\bar{\sigma}_{S,0}^2. \quad (70)$$

Rearranging, taking expectation, and summing over $k = 0, \dots, K$ yield

$$\begin{aligned}\frac{\mu_g}{4}\sum_{k=0}^K \gamma_k\,\mathbb{E}\|E_Z^k\|^2 \le{}& \mathbb{E}\|E_Z^0\|^2 + \frac{9B_{ZY}^2+\sigma_{S,2}^2}{2\mu_g}\sum_{k=0}^K \gamma_k\,\mathbb{E}\|E_Y^k\|^2 \\ &+ \frac{4L_{Z^*}^2}{\mu_g}\sum_{k=0}^K \frac{\alpha_k^2}{\gamma_k}\mathbb{E}\|x_+^k - x^k\|^2 + 2\bar{\sigma}_{S,0}^2\sum_{k=0}^K \gamma_k^2.\end{aligned} \quad (71)$$

Using the coupling $\gamma_k = c_2\alpha_k$ and the bound (50), we obtain

$$\sum_{k=0}^K \alpha_k\,\mathbb{E}\|E_Z^k\|^2 \le C_{Zx}\sum_{k=0}^K \alpha_k\,\mathbb{E}\|x_+^k - x^k\|^2 + C_{Z,0} + C_{Z,1}\sum_{k=0}^K \alpha_k^2,$$

where one may take

$$C_{Zx} := \frac{16L_{Z^*}^2}{c_2^2\mu_g^2} + \frac{2(9B_{ZY}^2 + \sigma_{S,2}^2)}{\mu_g^2} C_{Yx},$$

$$C_{Z,0} := \frac{4}{c_2\mu_g}\mathbb{E}\|E_Z^0\|^2 + \frac{2(9B_{ZY}^2 + \sigma_{S,2}^2)}{\mu_g^2} C_{Y,0},$$

$$C_{Z,1} := \frac{8c_2}{\mu_g}\bar{\sigma}_{S,0}^2 + \frac{2(9B_{ZY}^2 + \sigma_{S,2}^2)}{\mu_g^2} C_{Y,1}.$$

This proves (51). □

Lemma 7 (Hypergradient bias bound)**.** *Suppose that* (A1) *holds and define*

$$D_x(x, \boldsymbol{Y}, \boldsymbol{Z}) := \nabla_x f(x, \boldsymbol{Y}) - \mathcal{J}(x, \boldsymbol{Y})[\boldsymbol{Z}]. \tag{72}$$

Then for any $x \in \mathbb{R}^{d_x}$ and any $\boldsymbol{Y}, \boldsymbol{Z} \in \mathcal{Y}$,

$$\|D_x(x, \boldsymbol{Y}, \boldsymbol{Z}) - \nabla\Phi(x)\|^2 \le C_{xY}\|\boldsymbol{Y} - \boldsymbol{Y}^*(x)\|^2 + C_{xZ}\|\boldsymbol{Z} - \boldsymbol{Z}^*(x)\|^2, \tag{73}$$

where one may take

$$C_{xY} := 3\left(L_{f,1}^2 + \frac{L_{f,0}^2 L_{g,2}^2}{\mu_g^2}\right), \quad C_{xZ} := 3L_{g,1}^2. \tag{74}$$

Proof. Write $\boldsymbol{Y}^* := \boldsymbol{Y}^*(x)$ and $\boldsymbol{Z}^* := \boldsymbol{Z}^*(x)$. Then

$$\begin{aligned} D_x(x, \boldsymbol{Y}, \boldsymbol{Z}) - \nabla\Phi(x) = \big(\nabla_x f(x, \boldsymbol{Y}) - \nabla_x f(x, \boldsymbol{Y}^*)\big) - \mathcal{J}(x, \boldsymbol{Y})[\boldsymbol{Z} - \boldsymbol{Z}^*] \\ - \big(\mathcal{J}(x, \boldsymbol{Y}) - \mathcal{J}(x, \boldsymbol{Y}^*)\big)[\boldsymbol{Z}^*]. \end{aligned}$$

The first term is controlled by $L_{f,1}$, the second by (27), and the third by (28) together with (20). Applying $(a+b+c)^2 \le 3(a^2+b^2+c^2)$ yields the claimed constants. □

Lemma 8 (Variance of the hypergradient sample and a bound for $\|h^{k+1} - h^k\|$)**.** *Assume* (A1)–(A3) *and the sampling conditions in Lemma 5. Recall*

$$\widehat{W}^{k+1} := \tilde{U}_x^{k+1} - \tilde{\mathcal{J}}^{k+1}[\boldsymbol{P}^{k\top}\boldsymbol{Z}^k] \in \mathbb{R}^{d_x}.$$

Then there exist constants $\sigma_w^2, c_{wZ} \ge 0$ such that

$$\mathbb{E}[\|\widehat{W}^{k+1} - \mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k]\|^2] \le \sigma_w^2 + c_{wZ}\,\mathbb{E}\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2. \tag{75}$$

Moreover, if $h^{k+1} = (1-\theta_k)h^k + \theta_k\widehat{W}^{k+1}$, then

$$\mathbb{E}\|h^{k+1} - h^k\|^2 \le \sigma_{h,k}^2, \tag{76}$$

where

$$\begin{aligned} \sigma_{h,k}^2 := 2\theta_k^2\mathbb{E}\Big[\|h^k - \nabla\Phi(x^k)\|^2 + \|\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - \nabla\Phi(x^k)\|^2\Big] \\ + \theta_k^2\Big(\sigma_w^2 + c_{wZ}\mathbb{E}\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2\Big). \end{aligned} \tag{77}$$

Proof. Fix $k \ge 0$. Lemma 5 gives

$$\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] = \nabla_x f(x^k, \boldsymbol{Y}^k) - \mathcal{J}(x^k, \boldsymbol{Y}^k)[\boldsymbol{Z}^k].$$

Moreover, (34b) yields

$$\mathbb{E}[\|\widehat{W}^{k+1} - \mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k]\|^2 \mid \mathcal{F}_k] \le \tilde{\sigma}_{f,1}^2 + (\tilde{\sigma}_{g,2}^2 + \Lambda_Y)\|\boldsymbol{Z}^k\|^2.$$

Using the elementary inequality

$$\|\boldsymbol{Z}^k\|^2 \le 2\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2 + 2\|\boldsymbol{Z}^*(x^k)\|^2$$

and the uniform bound (20), we obtain

$$\|\boldsymbol{Z}^k\|^2 \le 2\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2 + \frac{2L_{f,0}^2}{\mu_g^2}.$$

Taking expectations gives (75) with

$$c_{wZ} := 2(\tilde{\sigma}_{g,2}^2 + \Lambda_Y), \quad \sigma_w^2 := \tilde{\sigma}_{f,1}^2 + c_{wZ}\frac{L_{f,0}^2}{\mu_g^2}.$$

For the increment bound, use

$$h^{k+1} - h^k = \theta_k(\widehat{W}^{k+1} - h^k).$$

Conditioning on $\mathcal{F}_k$ and using the zero mean of $\widehat{W}^{k+1} - \mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k]$ gives

$$\mathbb{E}[\|h^{k+1} - h^k\|^2 \mid \mathcal{F}_k] = \theta_k^2\|\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - h^k\|^2 + \theta_k^2\mathbb{E}[\|\widehat{W}^{k+1} - \mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k]\|^2 \mid \mathcal{F}_k].$$

Using

$$\|\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - h^k\|^2 \le 2\|h^k - \nabla\Phi(x^k)\|^2 + 2\|\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - \nabla\Phi(x^k)\|^2$$

and then taking expectations proves (76) and (77). □

Lemma 9 (Moving-average tracking bound for h^k)**.** *Suppose that* (A1)–(A3) *and the sampling conditions in Lemma 5 hold. Let $x_+^k := x^k - h^k$, so that $x^{k+1} - x^k = \alpha_k(x_+^k - x^k)$, and assume*

$$0 < \theta_k \le 1, \quad \theta_k = c_3\alpha_k$$

for some $c_3 > 0$. Then for any $K \ge 0$,

$$\sum_{k=0}^{K}\alpha_k\mathbb{E}\|h^k - \nabla\Phi(x^k)\|^2 \le \frac{1}{c_3}\mathbb{E}\|h^0 - \nabla\Phi(x^0)\|^2 + 2\sum_{k=0}^{K}\alpha_k\,\mathbb{E}\|\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - \nabla\Phi(x^k)\|^2 + \frac{2L_{\nabla\Phi}^2}{c_3^2}\sum_{k=0}^{K}\alpha_k\mathbb{E}\|x_+^k - x^k\|^2 + c_3c_{wZ}\sum_{k=0}^{K}\alpha_k^2\mathbb{E}\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2 + c_3\sigma_w^2\sum_{k=0}^{K}\alpha_k^2.$$

Proof. Fix $k \ge 0$ and define $e^k := h^k - \nabla\Phi(x^k)$. Using the update $h^{k+1} = (1-\theta_k)h^k + \theta_k\widehat{W}^{k+1}$ and adding/subtracting $\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k]$ and $\nabla\Phi(x^k)$, we obtain

$$e^{k+1} = (1-\theta_k)e^k + \theta_k\big(\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - \nabla\Phi(x^k)\big) + \big(\nabla\Phi(x^k) - \nabla\Phi(x^{k+1})\big) + \theta_k\big(\widehat{W}^{k+1} - \mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k]\big).$$

Conditioning on $\mathcal{F}_k$, the last term has zero mean and is orthogonal in expectation to the remaining $\mathcal{F}_k$-measurable terms. Hence

$$\mathbb{E}[\|e^{k+1}\|^2 \mid \mathcal{F}_k] = \Big\|(1-\theta_k)e^k + \theta_k(\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - \nabla\Phi(x^k)) + \nabla\Phi(x^k) - \nabla\Phi(x^{k+1})\Big\|^2 + \theta_k^2\mathbb{E}[\|\widehat{W}^{k+1} - \mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k]\|^2 \mid \mathcal{F}_k].$$

By convexity of $\|\cdot\|^2$ and then $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$,

$$\begin{aligned}\mathbb{E}[\|e^{k+1}\|^2 \mid \mathcal{F}_k] &\le (1-\theta_k)\|e^k\|^2 + \theta_k\left\|\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - \nabla\Phi(x^k) + \frac{1}{\theta_k}(\nabla\Phi(x^k) - \nabla\Phi(x^{k+1}))\right\|^2 \\ &\quad + \theta_k^2\mathbb{E}[\|\widehat{W}^{k+1} - \mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k]\|^2 \mid \mathcal{F}_k] \\ &\le (1-\theta_k)\|e^k\|^2 + 2\theta_k\|\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - \nabla\Phi(x^k)\|^2 + \frac{2}{\theta_k}\|\nabla\Phi(x^{k+1}) - \nabla\Phi(x^k)\|^2 \\ &\quad + \theta_k^2\mathbb{E}[\|\widehat{W}^{k+1} - \mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k]\|^2 \mid \mathcal{F}_k].\end{aligned}$$

By the $L_{\nabla\Phi}$-Lipschitz continuity of $\nabla\Phi$ and the identity $x^{k+1} - x^k = \alpha_k(x_+^k - x^k)$,

$$\|\nabla\Phi(x^{k+1}) - \nabla\Phi(x^k)\|^2 \le \alpha_k^2L_{\nabla\Phi}^2\|x_+^k - x^k\|^2.$$

Substituting this bound and using Lemma 8 give

$$\mathbb{E}\|e^{k+1}\|^2 \le (1-\theta_k)\mathbb{E}\|e^k\|^2 + 2\theta_k\mathbb{E}\|\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - \nabla\Phi(x^k)\|^2 + \frac{2\alpha_k^2 L_{\nabla\Phi}^2}{\theta_k}\mathbb{E}\|x_+^k - x^k\|^2 + \theta_k^2\sigma_w^2 + \theta_k^2 c_{wZ}\mathbb{E}\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2.$$

Rearranging and summing from $k=0$ to K yields

$$\sum_{k=0}^{K}\theta_k\,\mathbb{E}\|e^k\|^2 \le \mathbb{E}\|e^0\|^2 + 2\sum_{k=0}^{K}\theta_k\mathbb{E}\|\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - \nabla\Phi(x^k)\|^2 + 2L_{\nabla\Phi}^2\sum_{k=0}^{K}\frac{\alpha_k^2}{\theta_k}\mathbb{E}\|x_+^k - x^k\|^2 + \sigma_w^2\sum_{k=0}^{K}\theta_k^2 + c_{wZ}\sum_{k=0}^{K}\theta_k^2\mathbb{E}\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2.$$

Substituting $\theta_k = c_3\alpha_k$ and dividing by c_3 proves the claim. □

Lemma 10 (Primal progress: summability of $\|h^k\|^2$)**.** *Assume that $\nabla\Phi$ is $L_{\nabla\Phi}$-Lipschitz continuous and that $\Phi^* := \inf_{x\in\mathbb{R}^{d_x}}\Phi(x) > -\infty$. Consider the update $x^{k+1} = x^k - \alpha_k h^k$ and define $x_+^k := x^k - h^k$. Then $x^{k+1} - x^k = \alpha_k(x_+^k - x^k)$ and $\|x_+^k - x^k\| = \|h^k\|$. Moreover, if $\alpha_k \le \frac{1}{2L_{\nabla\Phi}}$ for all k, then for any integer $K \ge 0$,*

$$\sum_{k=0}^{K}\alpha_k\,\mathbb{E}\|x_+^k - x^k\|^2 \le 4\big(\Phi(x^0) - \Phi^*\big) + 2\sum_{k=0}^{K}\alpha_k\,\mathbb{E}\|h^k - \nabla\Phi(x^k)\|^2. \tag{78}$$

Proof. The identities

$$x^{k+1} - x^k = \alpha_k(x_+^k - x^k), \quad \|x_+^k - x^k\| = \|h^k\|$$

follow directly from the definition $x_+^k = x^k - h^k$ and the update $x^{k+1} = x^k - \alpha_k h^k$.

Fix $k \ge 0$. By the $L_{\nabla\Phi}$-smoothness of Φ,

$$\Phi(x^{k+1}) \le \Phi(x^k) + \langle\nabla\Phi(x^k), x^{k+1} - x^k\rangle + \frac{L_{\nabla\Phi}}{2}\|x^{k+1} - x^k\|^2.$$

Substituting $x^{k+1} - x^k = -\alpha_k h^k$ gives

$$\Phi(x^{k+1}) \le \Phi(x^k) - \alpha_k\langle\nabla\Phi(x^k), h^k\rangle + \frac{L_{\nabla\Phi}}{2}\alpha_k^2\|h^k\|^2.$$

Using $2\langle a,b\rangle = \|a\|^2 + \|b\|^2 - \|a-b\|^2$ with $a = \nabla\Phi(x^k)$ and $b = h^k$, we obtain

$$\langle\nabla\Phi(x^k), h^k\rangle \ge \frac{1}{2}\|h^k\|^2 - \frac{1}{2}\|h^k - \nabla\Phi(x^k)\|^2.$$

Therefore

$$\Phi(x^{k+1}) \le \Phi(x^k) - \frac{\alpha_k}{2}\|h^k\|^2 + \frac{\alpha_k}{2}\|h^k - \nabla\Phi(x^k)\|^2 + \frac{L_{\nabla\Phi}}{2}\alpha_k^2\|h^k\|^2.$$

Rearranging,

$$\frac{\alpha_k}{2}(1 - L_{\nabla\Phi}\alpha_k)\|h^k\|^2 \le \Phi(x^k) - \Phi(x^{k+1}) + \frac{\alpha_k}{2}\|h^k - \nabla\Phi(x^k)\|^2.$$

If $\alpha_k \le \frac{1}{2L_{\nabla\Phi}}$, then $1 - L_{\nabla\Phi}\alpha_k \ge 1/2$, and hence

$$\frac{\alpha_k}{4}\|h^k\|^2 \le \Phi(x^k) - \Phi(x^{k+1}) + \frac{\alpha_k}{2}\|h^k - \nabla\Phi(x^k)\|^2.$$

Taking expectations and summing over $k = 0, \dots, K$ yield

$$\frac{1}{4}\sum_{k=0}^{K}\alpha_k\,\mathbb{E}\|h^k\|^2 \le \Phi(x^0) - \Phi^* + \frac{1}{2}\sum_{k=0}^{K}\alpha_k\,\mathbb{E}\|h^k - \nabla\Phi(x^k)\|^2.$$

Multiplying by 4 and using $\|x_+^k - x^k\| = \|h^k\|$ proves (78). □

C.4 Main convergence theorem

Theorem 2 (Convergence rate of **BROS**). *Suppose that* (A1)–(A3) *hold and that the sampling conditions in Lemma 5 hold. Assume* $\Phi^* := \inf_{x\in\mathbb{R}^{d_x}} \Phi(x) > -\infty$. *Consider the unconstrained* **BROS** *updates*

$$\begin{aligned} x^{k+1} &= x^k - \alpha_k h^k, \\ \boldsymbol{Y}^{k+1} &= \boldsymbol{Y}^k - \beta_k \widehat{\boldsymbol{V}}^{k+1}, \\ \boldsymbol{Z}^{k+1} &= \boldsymbol{Z}^k - \gamma_k \widehat{\boldsymbol{S}}^{k+1}, \\ h^{k+1} &= (1-\theta_k)h^k + \theta_k \widehat{W}^{k+1}. \end{aligned}$$

Define

$$x_+^k := x^k - h^k, \quad V_k := \|x_+^k - x^k\|^2 + \|h^k - \nabla\Phi(x^k)\|^2.$$

If the stepsizes are constant and coupled as

$$\alpha_k \equiv \alpha, \quad \beta_k \equiv c_1\alpha, \quad \gamma_k \equiv c_2\alpha, \quad \theta_k \equiv c_3\alpha,$$

with $\alpha > 0$ *sufficiently small so that Lemmas 6, 9, and 10 apply, define*

$$\rho_0 := \frac{2L_{\nabla\Phi}^2}{c_3^2} + 2(C_{xY}C_{Yx} + C_{xZ}C_{Zx}). \tag{79}$$

Assume

$$\rho_0 \le 1/8 \tag{80}$$

and

$$\alpha \le \min\left\{1, \frac{1}{c_3}, \frac{1}{8c_3 c_{wZ} C_{Zx}}\right\}. \tag{81}$$

Then there exist finite constants $C_0, C_1 > 0$*, independent of* K*, such that*

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}V_k \le \frac{C_0}{K\alpha} + C_1\alpha. \tag{82}$$

In particular, if $\alpha = \Theta(K^{-1/2})$*, then*

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\|\nabla\Phi(x^k)\|^2 = \mathcal{O}(K^{-1/2}).$$

Moreover, if $R \sim \mathrm{Unif}\{0,\dots,K-1\}$ *is independent of the algorithmic randomness, then*

$$\mathbb{E}\|\nabla\Phi(x^R)\|^2 \le \frac{2}{K}\sum_{k=0}^{K-1}\mathbb{E}V_k = \mathcal{O}(K^{-1/2}), \tag{83}$$

so to ensure $\mathbb{E}\|\nabla\Phi(x^R)\|^2 \le \varepsilon$ *it suffices to take* $K = \mathcal{O}(\varepsilon^{-2})$.

Remark 2 (Feasibility of the coupling constants)**.** *The condition* (80) *is non-vacuous. Indeed, Lemma 6 gives*

$$C_{Yx} = \frac{8L_{Y^*}^2}{c_1^2\mu_g^2}, \quad C_{Zx} = \frac{16L_{Z^*}^2}{c_2^2\mu_g^2} + \frac{2(9B_{ZY}^2 + \sigma_{S,2}^2)}{\mu_g^2}C_{Yx},$$

so

$$\rho_0 = \frac{2L_{\nabla\Phi}^2}{c_3^2} + 2(C_{xY}C_{Yx} + C_{xZ}C_{Zx})$$

can be made arbitrarily small by choosing c_1, c_2, c_3 *sufficiently large. Hence there always exist finite coupling constants for which* (80) *holds.*

Proof. Let

$$\Delta_0 := \Phi(x^0) - \Phi^*, \quad S_x := \sum_{k=0}^{K-1} \alpha\,\mathbb{E}\|x_+^k - x^k\|^2, \quad S_h := \sum_{k=0}^{K-1}\alpha\,\mathbb{E}\|h^k - \nabla\Phi(x^k)\|^2.$$

Step 1: start from the moving-average tracking inequality. Lemma 9 yields

$$S_h \le \frac{1}{c_3}\mathbb{E}\|h^0 - \nabla\Phi(x^0)\|^2 + 2\sum_{k=0}^{K-1}\alpha\,\mathbb{E}\|\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - \nabla\Phi(x^k)\|^2 + \frac{2L_{\nabla\Phi}^2}{c_3^2}S_x + c_3c_{wZ}\sum_{k=0}^{K-1}\alpha^2\mathbb{E}\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2 + c_3\sigma_w^2K\alpha^2. \tag{84}$$

Step 2: bound the hypergradient bias term via $\boldsymbol{Y}/\boldsymbol{Z}$ tracking. By Lemma 5 and the definition (72),

$$\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] = D_x(x^k, \boldsymbol{Y}^k, \boldsymbol{Z}^k).$$

Lemma 7 therefore implies

$$\mathbb{E}\|\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - \nabla\Phi(x^k)\|^2 \le C_{xY}\mathbb{E}\|\boldsymbol{Y}^k - \boldsymbol{Y}^*(x^k)\|^2 + C_{xZ}\mathbb{E}\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2.$$

Summing with weight α and using Lemma 6 give

$$\sum_{k=0}^{K-1}\alpha\,\mathbb{E}\|\mathbb{E}[\widehat{W}^{k+1} \mid \mathcal{F}_k] - \nabla\Phi(x^k)\|^2 \le (C_{xY}C_{Yx} + C_{xZ}C_{Zx})S_x + B_0 + B_1K\alpha^2, \tag{85}$$

where

$$B_0 := C_{xY}C_{Y,0} + C_{xZ}C_{Z,0}, \quad B_1 := C_{xY}C_{Y,1} + C_{xZ}C_{Z,1}.$$

Step 3: control the $\alpha^2\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2$ term. Since $\alpha_k \equiv \alpha$,

$$\sum_{k=0}^{K-1}\alpha^2\,\mathbb{E}\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2 = \alpha\sum_{k=0}^{K-1}\alpha\,\mathbb{E}\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2.$$

Applying the second inequality in Lemma 6 yields

$$\sum_{k=0}^{K-1}\alpha^2\,\mathbb{E}\|\boldsymbol{Z}^k - \boldsymbol{Z}^*(x^k)\|^2 \le \alpha\big(C_{Zx}S_x + C_{Z,0} + C_{Z,1}K\alpha^2\big). \tag{86}$$

Since (81) includes $\alpha \le 1$, the last term on the right satisfies $K\alpha^3 \le K\alpha^2$.

Step 4: close the recursion using primal descent. Substituting (85) and (86) into (84) gives

$$S_h \le A_{h,0} + \big(\rho_0 + c_3c_{wZ}C_{Zx}\alpha\big)S_x + A_{h,1}K\alpha^2 + A_{h,2}\alpha, \tag{87}$$

where

$$A_{h,0} := \frac{1}{c_3}\mathbb{E}\|h^0 - \nabla\Phi(x^0)\|^2 + 2B_0,$$

$$A_{h,1} := 2B_1 + c_3c_{wZ}C_{Z,1} + c_3\sigma_w^2, \quad A_{h,2} := c_3c_{wZ}C_{Z,0}.$$

On the other hand, Lemma 10 gives

$$S_x \le 4\Delta_0 + 2S_h. \tag{88}$$

Combining (87) and (88) yields

$$S_h \le A_{h,0} + \big(\rho_0 + c_3c_{wZ}C_{Zx}\alpha\big)(4\Delta_0 + 2S_h) + A_{h,1}K\alpha^2 + A_{h,2}\alpha.$$

By (80) and (81),

$$\rho_0 + c_3c_{wZ}C_{Zx}\alpha \le \frac{1}{8} + \frac{1}{8} = \frac{1}{4},$$

and therefore

$$S_h \le A_{h,0} + \Delta_0 + \frac{1}{2}S_h + A_{h,1}K\alpha^2 + A_{h,2}\alpha.$$

Rearranging gives

$$S_h \le 2A_{h,0} + 2\Delta_0 + 2A_{h,1}K\alpha^2 + 2A_{h,2}\alpha. \tag{89}$$

Substituting (89) into (88) yields

$$S_x \le 4\Delta_0 + 4A_{h,0} + 4\Delta_0 + 4A_{h,1}K\alpha^2 + 4A_{h,2}\alpha. \tag{90}$$

Step 5: divide by $K\alpha$. Dividing (89) and (90) by $K\alpha$ shows that

$$\frac{S_h}{K\alpha} \le \frac{\widetilde{C}_{h,0}}{K\alpha} + \widetilde{C}_{h,1}\alpha + \frac{\widetilde{C}_{h,2}}{K} \quad \text{and} \quad \frac{S_x}{K\alpha} \le \frac{\widetilde{C}_{x,0}}{K\alpha} + \widetilde{C}_{x,1}\alpha + \frac{\widetilde{C}_{x,2}}{K},$$

for suitable finite constants $\widetilde{C}_{\cdot,\cdot}$ independent of K. Because $\alpha \le 1$, we have $1/K \le 1/(K\alpha)$, so the $1/K$ terms can be absorbed into the $1/(K\alpha)$ term. Summing the last two inequalities proves (82) for some finite constants $C_0, C_1 > 0$.

If $\alpha = \Theta(K^{-1/2})$, then both right-hand sides are $\mathcal{O}(K^{-1/2})$, which proves the displayed rate for

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\|h^k - \nabla\Phi(x^k)\|^2 \quad \text{and} \quad \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\|x_+^k - x^k\|^2.$$

Finally,

$$\|\nabla\Phi(x^k)\|^2 \le 2\|x_+^k - x^k\|^2 + 2\|h^k - \nabla\Phi(x^k)\|^2 = 2V_k,$$

which yields (83) and the $\mathcal{O}(\varepsilon^{-2})$ complexity claim. □

Corollary 1 (An explicit constant-stepsize choice)**.** *Under the assumptions of Theorem 2, fix any constants $c_1, c_2, c_3 > 0$ such that the step-size ratio conditions of Lemma 6 hold and* (80) *is satisfied; such constants exist by Remark 2. Let $\bar{\alpha} > 0$ be any sufficiently small constant so that all hypotheses of Theorem 2 hold whenever $\alpha \le \bar{\alpha}$. If we choose*

$$\alpha_k \equiv \alpha := \min\{\bar{\alpha}, 1/\sqrt{K}\}, \quad \beta_k \equiv c_1\alpha, \quad \gamma_k \equiv c_2\alpha, \quad \theta_k \equiv c_3\alpha,$$

then

$$\mathbb{E}\|\nabla\Phi(x^R)\|^2 \le 2\left(\frac{C_0}{K\alpha} + C_1\alpha\right).$$

In particular, if $K \ge 1/\bar{\alpha}^2$, then $\alpha = 1/\sqrt{K}$ and

$$\mathbb{E}\|\nabla\Phi(x^R)\|^2 \le \frac{2(C_0 + C_1)}{\sqrt{K}}.$$

Consequently, to guarantee $\mathbb{E}\|\nabla\Phi(x^R)\|^2 \le \varepsilon$ it suffices to take $K = \mathcal{O}(\varepsilon^{-2})$.

C.5 A quadratic counterexample for naive projected auxiliary HVPs

The correction in **BROS** is needed because a naive projected auxiliary HVP can change the fixed point of the auxiliary linear system. The following deterministic quadratic example isolates this issue. There is no stochastic objective noise and no lower-level tracking error; the only error comes from replacing the full HVP Hz by the uncorrected projected HVP $QHQz$.

Example 1 (Naive projected auxiliary HVP can give a nonstationary limit)**.** *Consider a deterministic bilevel problem with scalar upper variable $x \in \mathbb{R}$ and one lower-level matrix block $Y \in \mathbb{R}^{3\times 1}$, which we identify with a vector $y \in \mathbb{R}^3$:*

$$g(x, y) = \frac{1}{2}y^\top Hy - xc^\top y, \quad f(x, y) = \frac{1}{2}x^2 + d^\top y,$$

where $H = \mathrm{diag}(1, 2, 3)$ and $c = d = e_1 = (1, 0, 0)^\top$. The lower problem is strongly convex, and its solution is

$$y^*(x) = H^{-1}cx = xe_1.$$

The auxiliary linear system for the true hypergradient is

$$Hz^* = d, \quad z^* = H^{-1}d = e_1.$$

Moreover, since $\nabla^2_{xy}g[z] = -c^\top z$, the true hypergradient is

$$\nabla\Phi(x) = \nabla_x f(x, y^*(x)) - \nabla^2_{xy}g(x, y^*(x))[z^*] = x + c^\top z^* = x + 1.$$

Thus the unique stationary point of the true bilevel objective is $x^* = -1$. Now sample a scaled subspace lifting matrix

$$P = \sqrt{\frac{3}{2}}\, U_2 \in \mathbb{R}^{3\times 2}, \quad Q = PP^\top,$$

where U_2 contains the first two columns of a Haar-uniform orthogonal matrix in $O(3)$. Then $\mathbb{E}[Q] = I$, so the lifted lower gradient remains unbiased. However, the naive projected auxiliary HVP has mean $\bar{H}z := \mathbb{E}[QHQ]z$. By Lemma 3(ii), with $m = 3$ and $r = 2$, we have $a = \frac{21}{20}$ and $b = \frac{3}{20}$, and therefore

$$\bar{H} = aH + b\,\mathrm{tr}(H)I = \mathrm{diag}\left(\frac{39}{20}, 3, \frac{81}{20}\right).$$

Hence the mean field of the naive projected auxiliary recursion is

$$z^{k+1} = z^k - \gamma(\bar{H}z^k - d).$$

For any $0 < \gamma < 2/\lambda_{\max}(\bar{H})$, this recursion converges to $\bar{z} = \bar{H}^{-1}d = \frac{20}{39}e_1$, which differs from the true auxiliary solution $z^* = e_1$.

Even if the lower variable is kept exactly at $y^*(x)$, an upper update using this limiting biased auxiliary variable follows the direction

$$\bar{h}(x) = x + c^\top \bar{z} = x + \frac{20}{39}.$$

Thus the recursion

$$x^{k+1} = x^k - \eta\bar{h}(x^k)$$

converges, for any $0 < \eta < 2$, to $\bar{x} = -\frac{20}{39}$. However, this point is not stationary for the true bilevel objective, since

$$\nabla\Phi(\bar{x}) = \bar{x} + 1 = \frac{19}{39} \neq 0.$$

Consequently, for any $\varepsilon < \left(\frac{19}{39}\right)^2$, the naive projected auxiliary update cannot guarantee $\mathbb{E}\|\nabla\Phi(x)\|^2 \leq \varepsilon$ even in this noiseless quadratic problem.

This example shows that correcting the projected auxiliary HVP is necessary for targeting the original bilevel stationary point. The issue is not caused by stochastic variance or imperfect lower-level optimization; it is caused by the biased mean of the uncorrected projected auxiliary HVP.

D Memory analysis for Non-GQA Transformer blocks

D.1 Transformer layer and memory components

We follow the LLaMA-style Transformer layer used in the CR-Net memory discussion [31]. Let b be the micro-batch size, s the sequence length, n the hidden size, and h the number of attention heads. In the non-GQA case, the query, key, and value projections all have output width n. Folding the batch and sequence dimensions into the row dimension, a layer input can be viewed as $X_l \in \mathbb{R}^{bs\times n}$. Omitting layer normalization and RoPE, the attention part is

$$Y_l^Q := X_l W_l^Q, \quad Y_l^K := X_l W_l^K, \quad Y_l^V := X_l W_l^V,$$
$$\mathrm{Att}_l := \mathrm{softmax}\left(\frac{Y_l^Q (Y_l^K)^\top}{\sqrt{n/h}}\right) Y_l^V W_l^O + X_l.$$

With SwiGLU and intermediate width $m = \frac{8}{3}n$, the feed-forward part uses $W_l^g, W_l^u \in \mathbb{R}^{n\times m}$ and $W_l^d \in \mathbb{R}^{m\times n}$:

$$X_{l+1} := \left(\mathrm{SwiGLU}(\mathrm{Att}_l W_l^g) \odot (\mathrm{Att}_l W_l^u)\right) W_l^d + \mathrm{Att}_l.$$

We count scalar memory slots for one trainable decoder block and omit datatype constants. For a full-space method, the non-GQA trainable tuple is

$$\boldsymbol{Y} = (Y_{qkv}, Y_o, Y_{ff}, Y_d),$$

Table 3: Non-GQA lower-network peak-memory proxy for full-space exact recursion and **BROS** on one trainable LLaMA-style decoder block. The HVP/JVP column reports derivative-query scale rather than a backend-specific peak constant.

Method	Lower/aux. states	Activations	Grad./directions	HVP/JVP scale	Peak proxy
Full-space exact	$24n^2$	$15bsn + 2bhs^2$	$24n^2$	full-space HVP/JVP	$48n^2 + 15bsn + 2bhs^2$
BROS	$24n^2$	$\frac{28}{3}bsn + 2bhs^2 + 4bsr$	$\frac{62}{3}rn$	projected HVP/JVP	$24n^2 + \frac{28}{3}bsn + 2bhs^2 + 4bsr + \frac{62}{3}rn$

where $Y_{qkv} := [Y_q, Y_k, Y_v] \in \mathbb{R}^{n\times 3n}$, $Y_o \in \mathbb{R}^{n\times n}$, $Y_{ff} := [Y_g, Y_u] \in \mathbb{R}^{n\times \frac{16}{3}n}$, and $Y_d \in \mathbb{R}^{\frac{8}{3}n\times n}$. Hence

$$S_{\text{full}} := |\boldsymbol{Y}| = 3n^2\ (\text{QKV}) + n^2\ (\text{O}) + \frac{16}{3}n^2\ (\text{FFN up/gate}) + \frac{8}{3}n^2\ (\text{FFN down}) = 12n^2.$$

The trainable-side activations stored by a full-space backward pass have size

$$A_{\text{full}} := 15bsn\ (\text{saved hidden}) + 2bhs^2\ (\text{attention}),$$

where the $2bhs^2$ term comes from attention-score and attention-probability tensors.

D.2 BROS memory in the non-GQA case

For **BROS**, we use randomized projectors $P_{qkv}, P_o, P_{ff} \in \mathbb{R}^{n\times r}$ and $P_d \in \mathbb{R}^{\frac{8}{3}n\times r}$. The projected lower and auxiliary variables have the same block shapes

$$B_{qkv}, C_{qkv} \in \mathbb{R}^{r\times 3n}, \quad B_o, C_o \in \mathbb{R}^{r\times n}, \quad B_{ff}, C_{ff} \in \mathbb{R}^{r\times \frac{16}{3}n}, \quad B_d, C_d \in \mathbb{R}^{r\times n}.$$

Thus one projected tuple has size

$$S_{\text{proj}} := 3rn\ (\text{QKV}) + rn\ (\text{O}) + \frac{16}{3}rn\ (\text{FFN up/gate}) + rn\ (\text{FFN down}) = \frac{31}{3}rn.$$

BROS stores only random seeds for the projectors and reconstructs them when needed, so there is no persistent projector-storage term. The projected trainable-side activation footprint is

$$A_{\text{BROS}} := \frac{28}{3}bsn\ (\text{hidden}) + 2bhs^2\ (\text{attention}) + 4bsr\ (\text{projected inputs}).$$

The peak-memory proxy below focuses on trainable-side lower-network memory and omits upper-variable slots. We decompose the counted peak-memory proxy as

$$M_{\text{peak}} = M_{\text{state}}\ (\text{states}) + M_{\text{act}}\ (\text{activations}) + M_{\text{dir}}\ (\text{gradients/directions}).$$

Here M_{state} counts persistent lower and auxiliary variables, M_{act} counts saved trainable-side activations, and M_{dir} counts online lower-gradient and auxiliary-gradient directions. The HVP/JVP autodiff tapes are implementation-dependent, so we report their query scale separately rather than hiding them inside a fixed constant. For **BROS**, the theorem-aligned recursion stores the full lower/auxiliary iterates $(\boldsymbol{Y}^k, \boldsymbol{Z}^k)$, so $M_{\text{state}}^{\textbf{BROS}} = 2S_{\text{full}} = 24n^2$, while the projected lower and auxiliary gradient directions have size $M_{\text{dir}}^{\textbf{BROS}} = 2S_{\text{proj}} = \frac{62}{3}rn$. Therefore

$$M_{\text{peak}}^{\textbf{BROS}} = 24n^2\ (\text{state}) + \frac{28}{3}bsn\ (\text{hidden}) + 2bhs^2\ (\text{attn}) + 4bsr\ (\text{proj-act}) + \frac{62}{3}rn\ (\text{grad/dir}). \quad (91)$$

For the full-space exact single-loop baseline, the persistent lower/auxiliary states are the same, the trainable-side activations remain A_{full}, and the lower/auxiliary gradient directions contribute $2S_{\text{full}} = 24n^2$. Table 3 compares the resulting peak-memory proxy.

D.3 Memory of other bilevel baselines

We next place representative baselines from the references into the same non-GQA block model under the same peak-memory proxy. MA-SOBA keeps the lower iterate and an auxiliary linear-system iterate, and its lower/auxiliary gradient directions are full-space [7]. FdeHBO keeps the lower iterate y_t, the linear-system auxiliary iterate v_t, one-step-lag iterates, and full-space recursive momentum estimators for the lower and auxiliary recursions [63]. Penalty methods avoid the auxiliary linear system, so the favorable finite-penalty row only counts the lower state and full-space lower-gradient direction [42]. Bilevel-ZOFO is omitted because its memory is governed by a PEFT lower-level parameterization and a zeroth-order upper update, rather than the full trainable lower-network model considered in this table [51].

Table 4: Baseline lower-network peak-memory proxy under the same non-GQA decoder-block model. The HVP/JVP column reports derivative-query scale rather than a backend-specific peak constant.

Method	Lower/aux. states	Activations	Grad./directions	HVP/JVP scale	Peak proxy
MA-SOBA [7]	$24n^2$	$15bsn + 2bhs^2$	$24n^2$	full-space HVP/JVP	$48n^2 + 15bsn + 2bhs^2$
FdeHBO [63]	$48n^2$	$15bsn + 2bhs^2$	$24n^2$	full-space finite-diff. directions	$72n^2 + 15bsn + 2bhs^2$
Penalty [42]	$12n^2$	$15bsn + 2bhs^2$	$12n^2$	no auxiliary HVP/JVP	$24n^2 + 15bsn + 2bhs^2$
BROS	$24n^2$	$\frac{28}{3}bsn + 2bhs^2 + 4bsr$	$\frac{62}{3}rn$	projected HVP/JVP	$24n^2 + \frac{28}{3}bsn + 2bhs^2 + 4bsr + \frac{62}{3}rn$

Table 5: BROS updates by task and training stage.

Task/stage	x	$\boldsymbol{Y}$ / model	$\boldsymbol{Z}$	h
MNIST	Direct via h	Direct	Direct residual	EMA
CIFAR-10	Direct via h	Direct	Direct residual	EMA
280M proxy	Direct via h	Direct	Direct residual	EMA
280M main	Fixed mixture	AdamW	—	—
CIFAR-100 ViT	Adam	AdamW	Direct residual	EMA

Table 6: Model-configuration hyperparameters for MNIST hyper-data cleaning.

Lower model	Input dim.	Classes	Lower weights	Upper weights	Layers
Logistic regression	784	10	10×784	20000	1

E Additional experiments

Throughout this appendix, the experimental shorthand r/n denotes the chosen subspace rank r divided by the relevant layer width or hidden size n. In the theoretical notation, the corresponding projector ratio is r_ℓ/m_ℓ for each block Y_ℓ.

Table 5 lists the BROS updates by variable and training stage. "Direct" means no optimizer momentum or adaptive preconditioning; EMA denotes an exponential moving average. Baseline-specific settings are reported below.

E.1 MNIST hyper-data cleaning

We consider the convex MNIST hyper-data-cleaning problem [48], where the upper-level variable assigns sample weights and the lower-level problem trains a classifier on corrupted labels. The MNIST training split is divided into 20k noisy training examples and 5k clean validation examples, and the standard 10k clean test set is used for final evaluation. The training labels are corrupted by 50% random label noise, while validation and test labels remain clean. The lower model is a convex logistic-regression classifier with per-sample hyper-weights. All methods use the same data split, batch size 128, seed 42, 6000 outer steps, and evaluation interval 100 steps. All MNIST runs are conducted on a single NVIDIA A100 GPU (80 GB).

The common regularization and optimization settings are $C_r = 0.05$, gradient clipping 1.0, exact HVP/JVP evaluation, power learning-rate decay with inner decay 0.4 and outer decay 0.6, and $\delta_\epsilon = 0.01$ unless specified otherwise. MA-SOBA uses inner learning rate 0.1, outer learning rate 3.0, $\theta = 0.08$, z learning rate 0.02, and z clipping 1000. **BROS** uses inner learning rate 0.1, outer learning rate 2.0, $\delta_\epsilon = 0.005$, $\theta = 0.08$, z learning rate 0.02, z clipping 1000, and experimental rank ratio $r/n = 0.25$ in the main run. The rank-ratio ablation uses $r/n \in \{0.01, 0.05, 0.10, 0.25, 0.33, 0.50\}$. Penalty uses inner learning rate 0.06, outer learning rate 0.3, penalty coefficient 0.03, and inner clean coefficient 0.0; FdeHBO uses inner learning rate 0.06, outer learning rate 0.6, $\delta_\epsilon = 0.03$, and momentum 0.4; ZOFO uses inner learning rate 0.06, outer learning rate 0.05, perturbation scale 0.02, and gradient clipping 0.25.

Table 7: Model-configuration hyperparameters for data-mixture learning.

Parameters	Hidden	Intermediate	KV heads	Heads	Layers
280M	1024	4096	16	16	16

Table 8: Memory and per-step time for different algorithms in data-mixture learning.

Method	Peak memory	Memory vs. MA-SOBA	Per-step time (s)
DoReMi [58]	9447.0 MB	70.0%	0.149
MA-SOBA [7]	13496.9 MB	100.0%	0.278
Penalty [42]	10190.5 MB	75.5%	0.272
FdeHBO [63]	12896.3 MB	95.6%	0.260
ZOFO [51]	10047.1 MB	74.4%	0.556
BROS	7436.0 MB	55.1%	0.301

E.2 Data-mixture learning

We evaluate data-mixture learning on 17 copyright-free Pile domains. The experiment follows a proxy-to-main pipeline: a 280M GPT-style proxy model first learns domain weights, and a separate 280M main model is then trained from scratch using the learned mixture. The tokenizer is EleutherAI/gpt-neox-20b [4], the sequence length is 256, and the proxy, reference, and main models all use 16 layers, hidden size 1024, 16 attention heads, vocabulary size 50304, and dropout 0.0. The reference model is trained for 100000 steps with the initial data mixture and is used for DoReMi excess-loss evaluation and proxy-stage reference losses. Original Pile Mixture trains the same 280M main model from scratch with the original Pile-domain proportions rather than proxy-learned weights. All data-mixture runs are conducted on 8 NVIDIA RTX 4090 GPUs (24 GB each) with DDP.

Proxy training uses global batch size 128, micro-batch size 16, gradient accumulation 1, bfloat16 precision, gradient checkpointing, seed 42, warmup 2000 steps, and 100000 total steps. All bilevel data-mixture methods start from the uniform weight vector over the 17 domains and use inner learning rate 0.15, outer learning rate 0.01, $C_r = 5 \times 10^{-4}$, gradient clipping 5.0, smoothing constant 10^{-3}, minimum weight floor 5×10^{-5}, cosine schedule, and Neumann depth 2. MA-SOBA uses $\delta_\epsilon = 0.01$, z learning rate 0.02, z clipping 1000, $\theta = 0.7$, and $\theta_{\min} = 0.05$; **BROS** uses the same moving-average settings with experimental rank ratio $r/n = 0.25$. Penalty uses penalty coefficient 0.1 and $\delta_\epsilon = 0.001$, FdeHBO uses $\delta_\epsilon = 0.01$ and momentum 0.9, and ZOFO uses perturbation scale 0.02, one perturbation, and upper update interval 10. Main-model training is from scratch with the learned final domain weights, total steps 200000, global batch size 128, micro-batch size 16, learning rate 3×10^{-4}, minimum learning rate 3×10^{-5}, warmup ratio 0.06, weight decay 0.1, gradient clipping 1.0, and bfloat16 precision.

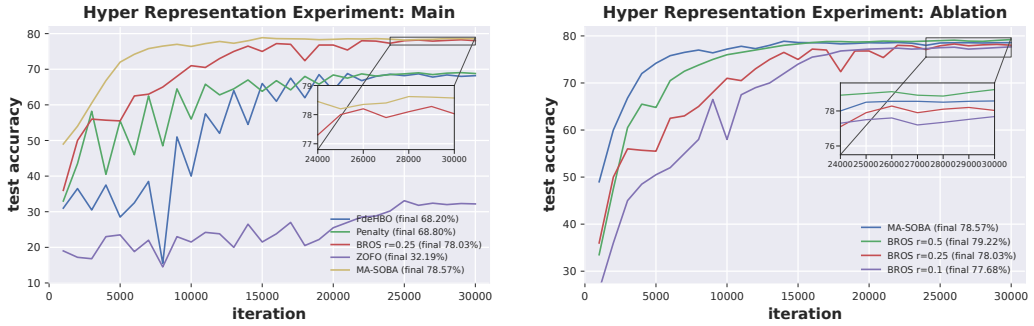
In addition to the main-model loss curves in Figure 2, we report proxy-stage system profiles for the data-mixture baselines. Table 8 summarizes the peak memory and average wall-clock time per proxy-training step. **BROS** has the smallest memory footprint among all compared methods, reducing peak memory to 55.1% of MA-SOBA while keeping the step time close to MA-SOBA, Penalty, and FdeHBO. Compared with ZOFO, which attains a similar final loss but relies on slower zeroth-order updates, **BROS** uses less memory and runs substantially faster per step. These results show that the randomized-subspace recursion improves memory efficiency without materially compromising proxy-stage training efficiency.

E.3 Hyper-representation learning

We evaluate CIFAR-10 hyper-representation learning [33], where the upper-level objective evaluates the representation and the lower-level problem trains a classifier under the current representation. The data split is 45k training examples, 5k validation examples, and 10k test examples. The upper-level variable is the ResNet-18 backbone trained from scratch, and the lower-level variable is its linear classification head. All methods are trained for 30000 steps with batch size 128, seed 42, CUDA with TF32 enabled, cosine decay, gradient clipping 20.0, and exact HVP/JVP evaluation; we record

Table 9: Model-configuration hyperparameters for CIFAR-10 hyper-representation learning.

Parameters	Input size	Stem width	Stage widths	Classes	Layers
≈ 11.2M	32×32	64	64/128/256/512	10	18



(a) Main comparison.

(b) Rank-ratio ablation.

Figure 3: **Hyper-representation learning.** CIFAR-10 test accuracy for main baselines and **BROS** rank-ratio ablations.

Table 10: CIFAR-10 hyper-representation results.

Method	Final step	Final test acc.	Best test acc.	Best step
FdeHBO [63]	30000	68.20%	68.71%	26000
Penalty [42]	30000	68.80%	69.05%	29500
MA-SOBA [7]	30000	78.57%	78.87%	15000
ZOFO [51]	30000	32.19%	33.09%	25000
BROS	30000	78.03%	78.28%	29000

wall-clock time and test accuracy at each evaluation point. All CIFAR-10 runs are conducted on a single NVIDIA A100 GPU (80 GB).

The common regularization is $C_r = 5 \times 10^{-4}$, with inner decay 0.4 and outer decay 0.6. MA-SOBA uses inner learning rate 0.4, outer learning rate 0.1, $\delta_\epsilon = 0.005$, $\theta = 0.1$, $\theta_{\min} = 0.05$, z learning rate 0.1, and z clipping 4000. **BROS** uses the same inner and outer learning rates, $\delta_\epsilon = 0.005$, $\theta = 0.1$, z learning rate 0.1, z clipping 4000, and experimental rank ratio $r/n = 0.25$ in the main run. The ablation compares $r/n \in \{0.50, 0.25, 0.10\}$. Penalty uses inner learning rate 0.08, outer learning rate 0.008, penalty coefficient 0.1, and $\delta_\epsilon = 0.01$; FdeHBO uses inner learning rate 0.1, outer learning rate 0.01, $\delta_\epsilon = 0.01$, and momentum 0.95; ZOFO uses inner learning rate 0.1, outer learning rate 0.02, and perturbation scale 0.02.

Figure 3 and Table 10 show that **BROS** with experimental rank ratio $r/n = 0.25$ obtains 78.03% final test accuracy and 78.28% best test accuracy, closely matching MA-SOBA (78.57% final, 78.87% best) and substantially outperforming FdeHBO, Penalty, and ZOFO. The rank-ratio ablation shows that $r/n = 0.50$ reaches 79.22%, while the more aggressive $r/n = 0.10$ still maintains 77.68% final accuracy.

E.4 CIFAR-100 ViT sample reweighting

We also evaluate a larger bilevel sample-reweighting task on CIFAR-100 with 20% label noise. The original 50k training images are split into 45k noisy training examples and 5k clean validation examples; the test set remains clean. The lower model is a small ViT trained from scratch with patch size 4, convolutional stem, embedding dimension 256, depth 8, and 4 heads. The upper-level objective selects sample weights using the clean validation split. All ViT runs are conducted on a single NVIDIA RTX PRO 6000 Blackwell GPU (96 GB).

Table 11: Model-configuration hyperparameters for CIFAR-100 ViT sample reweighting.

Parameters	**Patch size**	**Embed dim.**	**MLP dim.**	**Heads**	**Depth**
≈ 6.4M	4	256	1024	4	8

Table 12: CIFAR-100 ViT sample-reweighting results.

Method	Best step	Best val. acc.	Test acc. at best val.	Peak memory	Memory vs. MA-SOBA
Penalty [42]	26000	37.56%	37.04%	23.59 GB	99.9%
MA-SOBA [7]	4500	47.36%	48.07%	23.62 GB	100.0%
FdeHBO [63]	4500	48.04%	47.76%	23.16 GB	98.1%
ZOFO [51]	17000	42.68%	41.99%	11.20 GB	47.4%
BROS	6000	49.88%	49.05%	6.42 GB	27.2%

Table 13: Component timing on two GPUs. Entries give milliseconds and percentages normalized by the reported training-step times: 10.521 ms on RTX PRO 6000 GPU and 8.798 ms on A100 GPU. On RTX PRO 6000 GPU, component profiling is collected separately from training-step timing.

Component	RTX PRO 6000 ms/step (%)	A100 ms/step (%)
Haar-projector sampling	0.343 (3.26%)	0.116 (1.32%)
Projection and lifting	0.447 (4.25%)	0.328 (3.73%)
Bi-probe correction	0.987 (9.38%)	0.740 (8.41%)

All methods share the following training settings; the variable-wise BROS optimizers are specified in Table 5: batch size 512, model learning rate 10^{-3}, upper learning rate 10^{-4}, cosine decay, 1500 warmup steps, weight decay 0.08, and standard image augmentations including random crop, horizontal flip, RandAugment, random erasing, mixup/cutmix, and label smoothing. Low-rank methods use experimental rank ratio $r/n = 0.25$. Evaluation is performed every 500 steps. Model selection and early stopping use validation accuracy, with maximum 50000 steps, minimum 12000 steps, patience 6000, and minimum improvement 0.001.

Table 12 shows that **BROS** attains the best validation and test accuracy while using much less peak memory than full-space hypergradient baselines. In particular, **BROS** reaches 49.88% best validation accuracy and 49.05% test accuracy at the best-validation checkpoint, while using 6.42 GB peak memory compared with more than 23 GB for MA-SOBA, FdeHBO, and Penalty. ZOFO uses less memory than full-space baselines, but its test accuracy is substantially lower than **BROS**.

E.5 Bi-probe ablation

We compare naive projection-and-lift without bi-probes against its Rademacher-corrected counterpart on MNIST at label-noise rates 0.50, 0.65, and 0.80. Runs use 12000 steps, batch size 128, seed 42, and one NVIDIA RTX PRO 6000 GPU. We set $r/n = 0.025$, refresh the projector each step, and apply correction multiplier 2 from step 500. The trace-preserving anisotropic L_2 regularizer has mean coefficient 0.005, with 0.0005 on the central 20×20 pixels and bias and 0.0096992 elsewhere.

The corrected variant improves final validation accuracy over the naive variant by 0.46, 1.08, and 3.48 percentage points at the three noise rates, respectively.

E.6 Component and per-step timing

We profile projector sampling, projection/lift, and bi-probe correction on the MNIST data-cleaning task with batch size 128 and seed 42. Table 13 compares an NVIDIA RTX PRO 6000 GPU with an NVIDIA A100-80G GPU.

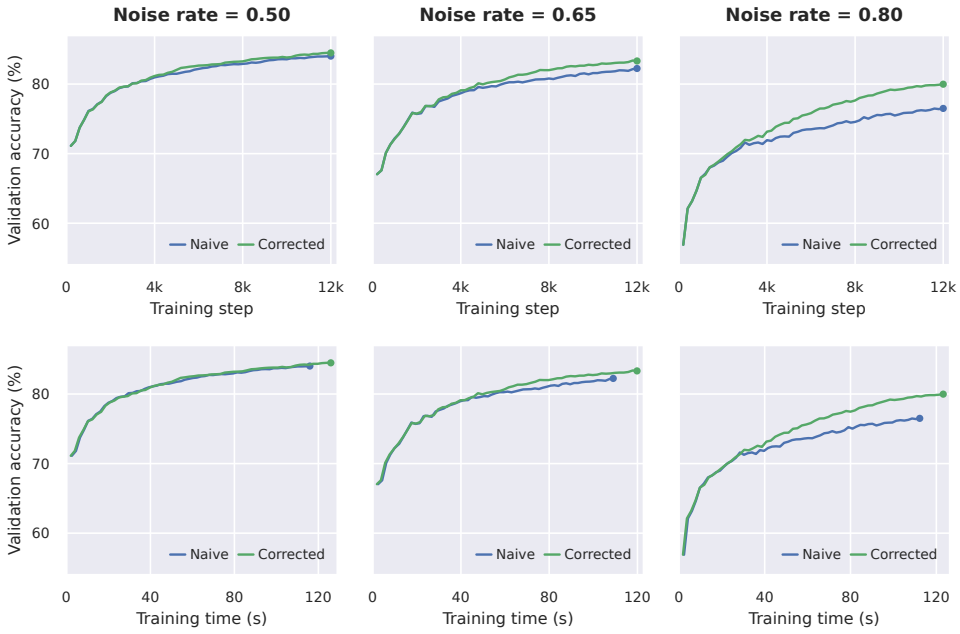


Figure 4: MNIST bi-probe ablation. Columns correspond to label-noise rates 0.50, 0.65, and 0.80. The top row plots validation accuracy against training steps; the bottom row uses training wall-clock time, excluding validation.

Table 14: Projection-and-probe timing ablation on A100.

Method	Per-step time (ms)	Time increase vs. MA-SOBA (%)
Full-space MA-SOBA	**7.170**	—
Naive projected-and-lift	7.694	$+7.31\%$
Naive projected-and-lift + Rademacher correction	8.798	$+22.70\%$

Table 15: MNIST test accuracy (%, higher is better) versus training wall-clock time.

Time (s)	MA-SOBA	Penalty	FdeHBO	ZOFO	BROS
13.85	82.61	80.96	80.98	80.95	82.58
26.39	83.56	82.05	82.34	82.06	83.38
38.92	84.00	82.87	82.91	82.69	83.72
51.46	84.44	83.37	83.25	83.28	84.06
63.99	84.45	83.60	83.56	83.56	84.58

Table 14 compares full-space MA-SOBA, naive projection-and-lift, and the corrected variant using 6000 MNIST steps on a single A100-80G GPU, with batch size 128 and seed 42. The naive variant includes projector sampling and projection/lift but omits the bi-probe queries and correction contractions. The correction adds 1.104 ms per step, or 14.35%, relative to the naive variant.

E.7 Performance under wall-clock budgets

Tables 15 and 16 compare performance at shared wall-clock budgets in the MNIST and 280M data-mixture settings. Each entry uses the nearest recorded checkpoint at or before the stated time. The 280M main model is evaluated every 5000 steps over a 200000-step run.

At 63.99 s, BROS reaches 84.58% MNIST test accuracy versus 84.45% for MA-SOBA. At 8 h, its 280M main-model loss is 2.8963 versus 2.9036 for MA-SOBA.

Table 16: 280M main-model average evaluation loss (lower is better) versus training wall-clock time. Original Pile Mixture uses the original domain proportions without proxy-learned weights.

Time (h)	Original Pile Mixture	MA-SOBA	DoReMi	Penalty	FdeHBO	ZOFO	BROS
4	3.1354	3.1330	3.3925	3.1128	3.1362	3.0975	3.1090
6	3.0487	3.0369	3.2955	3.0138	3.0224	3.0312	3.0265
8	2.9406	2.9036	3.1798	2.9334	2.9036	2.9063	2.8963

E.8 Licenses of used datasets and models

We use publicly available datasets, tokenizer assets, and model architectures for research experiments. We summarize their sources and associated licenses or terms below:

- **MNIST** [36]: The MNIST dataset is publicly available from the original MNIST database at `http://yann.lecun.com/exdb/mnist/`. The original public page does not explicitly state a separate dataset license; we use MNIST only as a standard research benchmark for hyper-data cleaning and do not redistribute it.
- **CIFAR-10 and CIFAR-100** [33]: The CIFAR datasets are publicly available from the University of Toronto CIFAR page at `https://www.cs.toronto.edu/~kriz/cifar.html`. The UCI Machine Learning Repository entry for CIFAR-10 lists the Creative Commons Attribution 4.0 International (CC BY 4.0) license; the original CIFAR-100 source page does not state a separate explicit license. We use both datasets only as standard research benchmarks and do not redistribute them.
- **Copyright-free Pile domains** [14]: The data-mixture experiment follows the copyright-free Pile-domain setting used in DoReMi [58]. We use the text domains only for research training and evaluation, do not redistribute the data, and follow the terms of the corresponding source components.
- **GPT-NeoX tokenizer** [4]: We use the tokenizer associated with EleutherAI/gpt-neox-20b, available from the Hugging Face model card. The model card lists the tokenizer and model assets under the Apache License 2.0.
- **ResNet and ViT architectures** [21, 11]: The CIFAR-10 and CIFAR-100 experiments train ResNet-18 and small ViT models from scratch. We do not use or release pretrained ResNet or ViT checkpoints.

F Broader impacts

This work focuses on improving the memory efficiency of stochastic bilevel optimization for neural and language-model training. By reducing the lower-level and auxiliary memory footprint of hypergradient computation, **BROS** may help run bilevel tuning, data-mixture learning, and sample-reweighting experiments on devices with lower memory capacity, thereby reducing resource consumption and carbon emissions associated with large-scale training. These efficiency gains may also make bilevel optimization techniques more accessible to smaller institutions and research teams operating under constrained compute budgets. Aiming to advance the field of bilevel optimization, we do not identify additional societal consequences that require further discussion here.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Refer to Abstract and section Introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Refer to section Conclusions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We present a theoretical insight for the observation, see Appendix for detailed proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Refer to Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The datasets we used are all open-access. We present the core code in the supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Refer to Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We show the variance or error bars of experimental results in experiments where we consider them essential.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix E reports the GPU type and memory capacity in each experimental setting, together with training steps and measured memory/time when available.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix F discusses the potential positive impacts of improved memory efficiency and notes that we do not identify additional societal consequences requiring further discussion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk data or models requiring responsible-release safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The appendix lists the existing datasets, tokenizer assets, and model architectures used in the experiments together with their sources and available licenses or terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The proposed algorithm can be used for LLM pre-training. Thus we present some experiments about LLM pre-training. Meanwhile, we use LLMs for editing.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.