Stochastic Gradient Methods with Online Scaling

Wanyu Zhang* Wenzhi Gao* Yinyu Ye* Madeleine Udell*

Abstract

This paper introduces Stochastic Online Scaled Gradient Methods (`SOSGM`), a generalization of the recently developed adaptive preconditioning framework in [15, 25] to stochastic optimization. Under standard assumptions, we establish convergence guarantees for `SOSGM` using large batchsize or variance reduction. `SOSGM` is compatible with popular diagonal and/or low-rank preconditioners as well as heavy-ball momentum, while maintaining memory and computation cost comparable to `Adam`. Extensive numerical experiments demonstrate the strong empirical performance of `SOSGM`. Using a diagonal preconditioner, `SOSGM` and its variants substantially outperform existing adaptive first-order methods across a range of statistical learning tasks.

1 Introduction

Optimization on large-scale datasets uses stochastic gradient descent (`SGD`) and its variants. However, `SGD` typically achieves only sublinear convergence rates on smooth, strongly convex objectives due to non-vanishing noise in the stochastic gradient. Although variance-reduction (VR) methods, such as `SVRG` [32], `SAGA` [18], and `Katyusha` [1], address this issue and offer improved convergence rates for convex problems, they still perform poorly on ill-conditioned data, which are pervasive in real-world problems [23, Table 2]. Ill-conditioning forces these methods to use conservative stepsizes for stability, limiting progress even with careful tuning.

Methods that use second-order information, such as Newton's method or `L-BFGS`, can converge quickly even on ill-conditioned data. While these methods do not scale to large datasets, many stochastic second-order methods have been proposed to deliver better performance than first-order methods. Most use subsampling-based approximations to the Hessian, which either directly compute the search direction using the inverse of a subsampled Hessian [9, 20, 46], or use subsampled Hessian to stabilize `L-BFGS`-style updates [11, 27, 43]. However, the former can be computationally heavy due to repeated linear-system solves involving subsampled Hessians, and both suffer from unstable curvature estimates.

Sketching-based preconditioners offer a complementary approach to handling ill conditioned problems. Notably, `SketchySGD` [23] and `PROMISE` [22] propose using scalable sketching methods to construct randomized low-rank preconditioners. These methods can boost the performance of `SVRG`, `SAGA`, and `Katyusha`, among others, and work well for large-scale dense data. However, for sparse data, the cost of storing and applying a low rank preconditioner compares poorly to the cost of storing or applying the data matrix, and so low-rank preconditioners are not recommended.

On a different front, in deep learning, adaptive gradient methods are widely used; many can be viewed as applying adaptive diagonal preconditioners, such as `AdaGrad` [21], `RMSProp` [30], and `Adam` [33]. These methods deliver faster convergence than `SGD` in practice. These methods are often strong empirically, but their update rules are typically heuristic and do not directly optimize a principled objective for choosing the preconditioner. While these methods accelerate training in practice, the diagonal preconditioners of adaptive optimizers like `Adam` are not designed to explicitly reduce the condition number of the problem.

Providing a theoretical foundation for stepsize adaptation, online scaled gradient methods (`OSGM`) [15, 25] offer a deterministic framework to adjust the stepsize / preconditioner, where the problem of choosing a stepsize is

*Stanford University. {zwanyu,gwz,yyye,udell}@stanford.edu

formulated as an online decision-making problem tackled by online learning algorithms. Theoretically, OSGM achieves convergence results that are asymptotically no worse than the optimal stepsize. By learning a matrix stepsize online, OSGM can adapt to the local geometry of the loss landscape and mitigate the effect of ill-conditioning.

The key missing piece is a stochastic counterpart with finite-sum convergence guarantees:

Can we design efficient adaptive stochastic gradient methods that are robust to ill-conditioning?

This work develops an affirmative answer to the question by introducing Stochastic Online Scaled Gradient Methods (SOSGM), which generalize OSGM to the stochastic optimization setting. SOSGM treats matrix stepsize selection as an online decision problem and include instantiations OSGM-SGD and OSGM-SVRG. SOSGM is compatible with scalar, diagonal, or matrix stepsizes, and optional heavy-ball momentum; for diagonal stepsizes, the memory and per-iteration cost are comparable to Adam.

Structure of the paper This paper is organized as follows. Section 2 introduces SOSGM, an OSGM-style framework to tune the matrix stepsize. Section 3 develops OSGM-SGD to learn the stepsize of SGD, and establishes high-probability convergence guarantees under a gradient-norm condition. Section 4 presents OSGM-SVRG, which adjusts the stepsize of SVRG in the outer loop. We prove the linear convergence for strongly convex objectives. Section 5 showcases empirical performance of SOSGM on statistical learning and deep learning problems.

Naming conventions Throughout the paper, colored, name-referenced algorithm names (such as OSGM-SGD) denote the theoretically analyzed SOSGM algorithms; uncolored, plain-text names (such as OSGM-SketchySVRG) denote the practical variants used in experiments. We use *stepsize* to refer to the stepsize/preconditioner in the optimization update, and we reserve *learning rate* for the stepsize used by the online gradient descent (hypergradient) update of the stepsize/preconditioner.

1.1 Related work

Hypergradient descent The hypergradient descent (HD) method was first presented in [3] as a heuristic to accelerate SGD. Similar ideas have been explored independently, such as incremental delta-bar-delta [53], stochastic meta-descent [48], and other adaptive schemes [31, 40]. This method was later rediscovered and called hypergradient descent by [6], which also extended the HD idea to tune SGD and Adam, with experiments on optimizing statistical learning problems and neural networks. Theoretical understanding of HD developed later, starting with [47] which analyzed convergence of HD for deterministic gradient descent on convex quadratic objectives, and continuing with [60] which analyzed convergence of a particular stochastic optimizer under the HD stepsize.

A more complete explanation for the empirical advantage of HD has recently been developed in a series of work [15, 16, 25, 26], which establishes the OSGM framework to choose a matrix stepsize by online learning. OSGM uses online gradient descent on different loss functions, and has strong trajectory-based convergence guarantees in the deterministic setting. This paper adapts OSGM for stochastic gradient methods.

Stochastic second-order methods A variety of stochastic analogs of classical Newton-type methods have been developed to address large-scale ill-conditioned optimization problems. Several works propose subsampling-based Hessian approximations, which either directly compute the search direction using the inverse of a subsampled Hessian [9, 20, 46], or use a subsampled Hessian to stabilize L-BFGS-style updates [11, 27, 43]. Several of these methods require expensive repeated linear-system solves involving subsampled Hessians, and most require a large batchsize or sufficiently good initialization for convergence guarantees.

Compared to stochastic second-order methods, SOSGM adapts to ill-conditioning without forming Hessian approximations or solving linear systems. Instead, SOSGM learns a matrix stepsize by online gradient descent, while retaining per-iteration costs comparable to Adam when instantiated with diagonal stepsizes.

Preconditioned stochastic gradient methods This line of work designs explicit preconditioners for stochastic gradient methods. [38] accelerated SVRG and Katyusha by applying inexact preconditioners derived from approximate Hessian solves. [22, 23] used randomized low-rank preconditioners to improve the convergence of SGD and methods such as SVRG, SAGA, and Loopless Katyusha (L-Katyusha [34]). They established global linear convergence with a constant batchsize, and demonstrated excellent performance through experiments on large-scale ill-conditioned machine learning problems. [52] showed how to use randomized low-rank preconditioners in the context of a stochastic proximal gradient methods, with impressive empirical results for regularized statistical learning problems like LASSO and elastic net.

Adaptive stochastic gradient methods Adaptive gradient methods are widely used in stochastic optimization, particularly for training neural networks. Many of these methods can be viewed as employing a diagonal preconditioner learned from previous gradients, such as AdaGrad [21], Adam [33], AdaHessian [61], Lion [14], Sophia [37], and Adafactor [50]. More recently, optimizers like Shampoo [29], K-FAC [42], and SOAP [59] apply structured (non-diagonal) preconditioners and show promising performance in training large language models.

Theoretical work on adaptive stochastic gradient methods develops principled stepsize selection procedures with provable convergence. For example, [41] proposed an adaptive stepsize based on local curvature estimates, with convergence rates depending on local geometry; its counterpart for stochastic gradient methods was studied in [5]. Line-search rules for SGD were explored in [56, 57], and [56] further established faster convergence than standard SGD under interpolation-type conditions. Finally, [39] analyzed the Polyak stepsize for SGD and provided convergence guarantees in both convex and non-convex settings. Recently, adaptive methods without knowing the problem parameters have been proposed, including D-adaptation for unknown distance-to-optimality [17] and schedule-free optimizers that avoid dependence on a pre-specified training horizon [19]. However, most work in this line focuses on *scalar* stepsizes and therefore does not improve the condition number of the problem.

1.2 Notations

We use $\|\cdot\|$ to denote the Euclidean norm of vectors or the operator norm of matrices, and $\langle\cdot,\cdot\rangle$ to denote the Euclidean or Frobenius inner product. The notation $\|A\|_F := \sqrt{\sum_{ij} a_{ij}^2}$ denotes the matrix Frobenius norm. Given a closed convex set $\mathcal{P}$, $\Pi_{\mathcal{P}}[\cdot]$ denotes the orthogonal projection onto $\mathcal{P}$; $\mathrm{dist}(P,\mathcal{P}) := \|P - \Pi_{\mathcal{P}}[P]\|_F$ denotes the distance between a point P and set $\mathcal{P}$; Given a vector $v \in \mathbb{R}^d$, $\mathrm{Diag}(v)$ denotes the diagonal matrix with elements of v on its diagonal. We use $\mathcal{X}^\star = \{x : f(x) = f^\star\}$ to denote the optimal set of f; $\mathrm{diam}(\mathcal{P}) = \max_{X,Y\in\mathcal{P}} \|X - Y\|_F$ denotes the diameter of the set $\mathcal{P}$ in Frobenius norm. A function f is L-smooth if it satisfies $\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|$ for all $x, y \in \mathbb{R}^d$. We use superscripts x^k to index algorithm iterates and subscripts P_k to index the stepsize sequence. The notation $\mathbb{S}^n$ denotes the set of n by n symmetric matrices. For asymptotic complexity, $\tilde{\mathcal{O}}(\cdot)$ hides polylogarithmic factors.

2 Stochastic online scaled gradient methods

To motivate the stochastic setting, this section first reviews the deterministic OSGM framework and the challenges that arise when gradients are noisy, then introduces SOSGM, a general framework for learning the matrix stepsize of stochastic gradient methods.

2.1 Deterministic online scaled gradient methods

Consider the unconstrained minimization of a deterministic smooth convex function $\min_{x\in\mathbb{R}^d} f(x)$ with preconditioned gradient descent:

$$x^{k+1} = x^k - P_k \nabla f(x^k),$$

where $P_k \in \mathbb{R}^{d\times d}$ is a matrix stepsize that can be scalar ($P_k = \alpha_k I$ for $\alpha_k \in \mathbb{R}$), diagonal ($P_k = \mathrm{diag}(v_k)$ for $v_k \in \mathbb{R}^d$) or a general matrix. `OSGM` [15, 25] is a framework that uses online learning to adjust stepsize $\{P_k\}$.

To motivate `OSGM`, consider the standard analysis of a linearly convergent method. The usual goal is to prove a uniform one-step contraction, $\frac{f(x^{k+1})-f^\star}{f(x^k)-f^\star} \le 1 - \frac{1}{\kappa}$, for all k. Multiplying these per-iteration bounds gives

$$\frac{f(x^K)-f^\star}{f(x^0)-f^\star} = \prod_{k=0}^{K-1} \frac{f(x^{k+1})-f^\star}{f(x^k)-f^\star} \le (1-\tfrac{1}{\kappa})^K.$$

Different from the above analysis, `OSGM` first chains the progress and gives an upper bound by the arithmetic-geometric mean inequality

$$\frac{f(x^K)-f^\star}{f(x^0)-f^\star} = \prod_{k=0}^{K-1} \frac{f(x^{k+1})-f^\star}{f(x^k)-f^\star} \le (\tfrac{1}{K}\sum_{k=0}^{K-1} \frac{f(x^{k+1})-f^\star}{f(x^k)-f^\star})^K.$$

From this, faster convergence is achieved by minimizing the average contraction ratio. Define $r_{x^k}(P_k) := \frac{f(x^k - P_k\nabla f(x^k))-f^\star}{f(x^k)-f^\star}$; it suffices to choose the stepsizes $\{P_k\}$ sequentially to minimize $\frac{1}{K}\sum_{k=1}^K r_{x^k}(P_k)$. In this view, stepsize selection is formulated as an online decision-making problem, thus motivating the use of online learning algorithms to adjust $\{P_k\}$. For example, online gradient descent

$$P_{k+1} = P_k - \eta \nabla r_{x^k}(P_k)$$

yields sublinear regret $\frac{1}{K}\sum_{k=1}^K r_{x^k}(P_k) \le \frac{1}{K}\sum_{k=1}^K r_{x^k}(\hat{P}) + \mathcal{O}(\frac{1}{\sqrt{K}})$ for any fixed stepsize $\hat{P}$ [45]. Choose $\hat{P}$ as a good stepsize $P_\star$ that achieves condition number $\kappa_\star < \kappa$ and progress $r_{x^k}(P_\star) \le 1 - \frac{1}{\kappa_\star}$ for all k. The convergence guarantee follows:

$$\frac{f(x^K)-f^\star}{f(x^0)-f^\star} \le (\tfrac{1}{K}\sum_{k=0}^{K-1} r_{x^k}(P_k))^K \le (1 - \tfrac{1}{\kappa_\star} + \mathcal{O}(\tfrac{1}{\sqrt{K}}))^K.$$

This result suggests that the performance of `OSGM` is competitive with a good stepsize $P_\star$ when the total number of iterations K is large, even without knowledge of $P_\star$.

The above algorithmic intuition is generalized into the `OSGM` framework. In each iteration, 1) stepsize scheduler makes decision P_k from a candidate set $\mathcal{P}$ and proposes an update $x^{k+1/2} = x^k - P_k\nabla f(x^k)$; 2) the landscape chooses the next iterate $x^{k+1} = \mathcal{M}(x^k, x^{k+1/2})$, for example, with a null step

$$x^{k+1} = \operatorname*{arg\,min}_{x\in\{x^k, x^{k+1/2}\}} \{f(x^k - P_k\nabla f(x^k)), f(x^k)\}, \qquad \text{(Null step)}$$

and provides feedback $\ell_{x^k}(P_k)$ to the scheduler. Two useful feedback functions are

$$\text{ratio}\ \ r_{x^k}(P_k) = \frac{f(x^k - P_k\nabla f(x^k))-f^\star}{f(x^k)-f^\star} \ \text{ or hypergradient}\ \ h_{x^k}(P_k) = \frac{f(x^k - P_k\nabla f(x^k))-f(x^k)}{\|\nabla f(x^k)\|^2}; \qquad (1)$$

3) scheduler updates the stepsize P_k by an online learning algorithm (such as online gradient descent) with respect to feedback $\ell_{x^k}(P_k)$.

2.2 Stochastic `OSGM`

This paper considers the following finite-sum problem,

$$\min_{x\in\mathbb{R}^d} f(x) := \tfrac{1}{n}\sum_{i=1}^n f_i(x). \qquad (2)$$

To solve (2), consider stochastic preconditioned gradient descent with a heavy-ball momentum (HBM),

$$x^{k+1} = x^k - P_k\, g^k + \beta_k(x^k - x^{k-1}),$$

where g^k is an unbiased estimator of the full gradient $\nabla f(x^k)$. For example, g^k could be

(SGD) the average of a batch of stochastic gradients, $\frac{1}{|\xi_k|}\sum_{i\in\xi_k}\nabla f_i(x^k)$, or

(SVRG) a VR estimator with a snapshot $\tilde{x}$, $\nabla f_i(x^k) - \nabla f_i(\tilde{x}) + \nabla f(\tilde{x})$.

We develop a framework, SOSGM (Algorithm 1), which adapts OSGM to learn matrix stepsizes for stochastic methods such as SVRG and SGD. At each iteration, SOSGM alternates between updating the iterate x^k and updating the stepsize P_k. This framework can be instantiated as the algorithms introduced later:

- OSGM-SGD. Update P_k at every iteration using stochastic ratio or hypergradient feedback.
- OSGM-SVRG. Update P_k every m inner iterations. Set $\ell_k = 0$ if $\mathrm{mod}(k, m) \neq 0$; otherwise, use deterministic regularized ratio or hypergradient feedback, since the full gradient is available.

Algorithm 1 SOSGM

1: **Input:** Initial point x^0, initial stepsize P_0, candidate stepsize set $\mathcal{P}$, OSGM learning rate η, momentum sequence $\{\beta_k\}$, feedback ℓ_k
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute stochastic gradient estimator g^k on x^k
4: Stochastic gradient step $x^{k+1} = x^k - P_k\, g^k + \beta_k(x^k - x^{k-1})$
5: Construct feedback ℓ_k and update stepsize P_k by OSGM $P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta\nabla\ell_k(P_k)]$
6: **end for**

2.2.1 Assumptions

To analyze the performance of these methods, the following assumptions are used throughout the paper.

Assumption 2.1. For each i, f_i is convex and L-smooth and μ-strongly convex with $\mu \geq 0$:

$$f_i(y) \geq f_i(x) + \langle \nabla f_i(x), y - x\rangle + \tfrac{\mu}{2}\|y - x\|^2, \quad \forall x, y.$$

This assumption holds for problems such as regularized generalized linear models.

The stepsize is chosen from a closed convex candidate set $\mathcal{P} \subseteq \mathbb{R}^{d\times d}$. It is natural to have the set of preconditioners include 0, as this represents a safe initialization that (we hope will learn a better preconditioner but) does not move the iterate, and also the theoretically optimal scalar stepsize $\frac{1}{L}I \in \mathcal{P}$. Hence the diameter of $\mathcal{P}$ must be at least $\frac{1}{L}$. We further impose the assumption that the diameter of $\mathcal{P}$ is in the order of $\frac{1}{L}$.

Assumption 2.2. The candidate set of stepsize $\mathcal{P}$ is bounded, $\mathrm{diam}(\mathcal{P}) \leq D$ and $D = \mathcal{O}(\frac{1}{L})$.

2.2.2 Failure of naive feedback design

In stochastic setting where the accurate function values and gradients are expansive or unavailable, one challenge of applying SOSGM is the noisy feedbacks. We first notice that a naive extension of hypergradient or ratio feedback cannot guarantee convergence.

For SGD, at iteration k, we sample a mini-batch ξ^k, take a stochastic gradient step on the sampled objective f_{ξ^k}. To apply SOSGM, define the feedback by measuring progress on the same sampled objective, i.e., $f_{\xi^k}(x) := \frac{1}{|\xi^k|}\sum_{i\in\xi^k} f_i(x)$.

$$r_{x^k,\xi^k}(P_k) = \frac{f_{\xi^k}(x^k - P_k\nabla f_{\xi^k}(x^k)) - f^\star}{f_{\xi^k}(x^k) - f^\star}, \quad h_{x^k,\xi^k}(P_k) = \frac{f_{\xi^k}(x^k - P_k\nabla f_{\xi^k}(x^k)) - f_{\xi^k}(x^k)}{\|\nabla f_{\xi^k}(x^k)\|^2}. \tag{3}$$

These feedbacks are *in-sample*: they reuse the same mini-batch ξ^k for both the update and the evaluation of progress. As a result, they can overfit to the selected mini-batch and fail to reflect progress on the full objective f.

We provide a counterexample in Section A, which shows that SOSGM with feedback (3) does not necessarily converge, even when each f_{ξ^k} is convex and smooth, all sampled objectives share the same minimizer, and f has bounded sublevel sets.

In the counterexample, the learned stepsize and resulting update are (locally) optimal for each sampled objective f_{ξ^k}, while still being suboptimal for f. This suboptimality with respect to f is not captured by the naive feedbacks (3), which evaluate progress only on the selected mini-batch.

We propose two strategies to mitigate the challenge of noisy feedback: out-of-sample feedbacks with large batchsizes for OSGM-SGD (Section 3), and full gradient update in the outerloop for OSGM-SVRG (Section 4).

3 SOSGM with large batchsize

In this section, we consider using OSGM to tune the stepsize of SGD,

$$x^{k+1} = x^k - P_k \nabla f_{\xi^k}(x^k).$$

A key difficulty is that naive stochastic extensions of the deterministic feedback functions can fail to converge (see Example A.1); out-of-sample feedback resolves this issue. We define out-of-sample feedbacks and present OSGM-SGD, which has convergence guarantees under the gradient norm condition and large batchsize.

3.1 OSGM-SGD

Define the out-of-sample feedbacks by using an independent sample ζ^k, which provides an unbiased estimator for function value decrease:

$$r_{x^k,\xi^k,\zeta^k}(P_k) = \frac{f_{\zeta^k}(x^k - P_k \nabla f_{\xi^k}(x^k)) - f^\star}{f_{\xi^k}(x^k) - f^\star}, \quad h_{x^k,\xi^k,\zeta^k}(P_k) = \frac{f_{\zeta^k}(x^k - P_k \nabla f_{\xi^k}(x^k)) - f_{\zeta^k}(x^k)}{\|\nabla f_{\xi^k}(x^k)\|^2}. \tag{4}$$

With this feedback design, we define OSGM-SGD in Algorithm 2.

Algorithm 2 OSGM-SGD

Input: Initial iterate x^0, initial stepsize P_0, candidate stepsize set $\mathcal{P}$, learning rate η, and feedback $\ell_{x,\xi,\zeta} \in \{r_{x,\xi,\zeta},\, h_{x,\xi,\zeta}\}$ defined by (4)
for $k = 0, 1, 2, \ldots$ **do**
  Sample ξ^k and ζ^k uniformly and independently
  $x^{k+1/2} = x^k - P_k \nabla f_{\xi^k}(x^k)$
  $x^{k+1} = \begin{cases} \arg\min_{x \in \{x^k,\, x^{k+1/2}\}} f_{\zeta^k}(x), & \text{if use } h_{x,\xi,\zeta}, \\ x^{k+1/2}, & \text{otherwise.} \end{cases}$
  $P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta \nabla \ell^k_{x^k,\xi^k,\zeta^k}(P_k)]$
end for

In this section, we establish convergence results for OSGM-SGD. Specifically, we consider two settings: 1) with access to the exact function value, and 2) with a noisy function value oracle (Assumption 3.2). The first setting is justified by derivative-free optimization, where an exact function value oracle is available and the gradient is estimated by finite difference.

We use the following assumption about the stochastic gradient and function value oracles throughout this section.

Assumption 3.1 (Gradient norm condition)**.** The stochastic gradient oracle $\nabla f_\xi(x)$ satisfies, $\mathbb{E}_\xi[\nabla f_\xi(x)] = \nabla f(x)$, and for any iterate $x \in \{x^k\}_{k=1,2,\ldots}$ and any $t > 0$,

$$\mathbb{P}\{\, \|\nabla f_\xi(x) - \nabla f(x)\| \geq t\, \|\nabla f(x)\|\} \leq 2\exp\big(-\tfrac{t^2}{2\sigma_1^2}\big). \tag{5}$$

where $\sigma_1 > 0$ controls the relative noise level. For a mini-batch of size b, the constant scales as $\sigma_1/\sqrt{b}$.

Remark 3.1. The gradient norm condition was first proposed in [12] to analyze the stochastic trust region method, and is a mainstay of subsequent literature [7, 10]. In the derivative-free optimization setting, where stochastic gradient estimates are obtained by sampling, the norm condition can be shown to hold with high probability [8].

When the exact function value is not available, we use the following assumption on the function value oracle.

Assumption 3.2. The stochastic function value oracle $f_\xi(x)$ satisfies, $\mathbb{E}_\xi[f_\xi(x)]=f(x)$, and for any iterate $x \in \{x^k\}_{k=1,2,\ldots}$ and any $t > 0$,

$$\mathbb{P}\{|f_\xi(x) - f(x)| \;\geq\; t\,|f(x) - f^\star|\} \;\leq\; 2\exp\Big(-\tfrac{t^2}{2\sigma_0^2}\Big)\,, \tag{6}$$

where $\sigma_0 > 0$ controls the relative noise level in the function value. We use $\sigma_0 = 0$ to denote the exact function value oracle. For a mini-batch of size b, the constant scales as $\sigma_0/\sqrt{b}$.

We provide an example where Assumptions 3.1 and 3.2 hold.

Example 3.2 (Least-squares with interpolation and sub-Gaussian features)**.** Suppose

$$f(x) = \tfrac{1}{n}\textstyle\sum_{i=1}^n f_i(x), \qquad f_i(x) = \tfrac{1}{2}(a_i^\top x - b_i)^2,$$

and assume an interpolation setting where $a_i^\top x^\star = b_i$ for all i, and a_i are sub-Gaussian with parameter σ_a. Let $H := \frac{1}{n}\sum_{i=1}^n a_i a_i^\top$, then Assumptions 3.1 and 3.2 hold with $\sigma_0 = \Theta(\sigma_a^2/\lambda_{\min}(H))$ and $\sigma_1 = \Theta(\sigma_a^2/\lambda_{\min}(H))$.

3.2 Convergence results

We have the following convergence guarantee for OSGM-SGD with feedback functions (4). Table 1 summarizes the theoretical results in this section.

Noise oracle	Function class	Feedback	Iteration complexity	Batchsize
Assumption 3.1, exact function value	L-smooth, μ-strongly convex	Ratio	$\tilde{\mathcal{O}}\big((1+\sigma_1^2)\kappa_\star \log \frac{1}{\varepsilon}\big)$	$\tilde{\mathcal{O}}(1)$
	L-smooth, μ-strongly convex	Hypergradient	$\tilde{\mathcal{O}}\big((1+\sigma_1^2)\kappa \log \frac{1}{\varepsilon}\big)$	$\tilde{\mathcal{O}}(1)$
	L-smooth, convex	Hypergradient	$\tilde{\mathcal{O}}\big(\frac{L\Delta^2(1+\sigma_1^2)}{\varepsilon}\big)$	$\tilde{\mathcal{O}}(1)$
Assumptions 3.1 and 3.2	L-smooth, μ-strongly convex	Ratio	$\tilde{\mathcal{O}}\big((1+\sigma_1^2)\kappa_\star \log \frac{1}{\varepsilon}\big)$	$\tilde{\mathcal{O}}(\kappa_\star^2)$
	L-smooth, μ-strongly convex	Hypergradient	$\tilde{\mathcal{O}}\big((1+\sigma_1^2)^2\kappa \log \frac{1}{\varepsilon}\big)$	$\tilde{\mathcal{O}}(\kappa^2)$

Table 1: Summary of Theoretical Results of OSGM-SGD (Theorem 3.3).

Theorem 3.3 (Convergence of OSGM-SGD)**.** Under Assumptions 3.1 and 3.2, suppose we run OSGM-SGD for K iterations. For any $\delta \in (0,1)$, define $\gamma(K,\delta) = 4\sqrt{\max\big\{\log\frac{4K}{\delta}, 1\big\}}$ and let $\tilde{\mathcal{O}}$ hides polynomial logarithmic terms of K and $1/\delta$. Then with probability $\geq 1-\delta$ we have the following convergence results.

(i) *Ratio feedback, strongly convex.* Suppose $\sigma_1 < \frac{1}{2\gamma(K,\delta)}$ and $\sigma_0 = \tilde{\mathcal{O}}(\frac{1}{L^2D^2\kappa_\star})$.

$$\tfrac{f(x^K)-f(x^\star)}{f(x^0)-f(x^\star)} \leq \big(1 - \tfrac{1}{2(1+\gamma(K,\delta)^2\sigma_1^2)\kappa_\star} + \tilde{\mathcal{O}}\big(\tfrac{1}{\sqrt{K}}\big)\big)^K.$$

(ii) *Hypergradient feedback, strongly convex.* Suppose $\sigma_1 = \tilde{\mathcal{O}}\left(\frac{1}{L^2D^2}\right)$ and $\sigma_0 = \tilde{\mathcal{O}}\left(\frac{1}{L^2D^2\kappa}\right)$.

$$\frac{f(x^K)-f(x^\star)}{f(x^0)-f(x^\star)} \leq \left(1 - \frac{1}{2(1+\sigma_1^2)\gamma(K,\delta)^2\kappa} + \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{K}}\right)\right)^K.$$

(iii) *Hypergradient feedback, convex.* Suppose $\sigma_1 = \tilde{\mathcal{O}}\left(\frac{1}{L^2D^2}\right)$ and $\sigma_0 = 0$.

$$f(x^K) - f(x^\star) \leq \min\left\{\frac{\Delta^2}{\max\left\{K\left(\frac{1}{4L(1+\gamma(K,\delta)^2\sigma_1^2)} - \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{K}}\right)\right), 0\right\}}, f(x^0) - f^\star\right\}.$$

Theorem 3.3 suggests that when the stochastic noise is small, the asymptotic convergence of OSGM-SGD is similar to the deterministic setting.

The requirement on the noise level σ_0, σ_1 can always be satisfied by using a sufficiently large batchsize. We distinguish the following two settings, when $\sigma_0 = 0$ and $\sigma_0 > 0$. If the function value oracle is exact, $\sigma_0 = 0$, Theorem 3.3 guarantees linear convergence as long as $\sigma_1 = \mathcal{O}(\frac{1}{L^2D^2})$ for both ratio and hypergradient feedback. Therefore, the batchsize needed is $\tilde{\mathcal{O}}(1)$, as $D = \mathcal{O}(1/L)$ by Assumption 2.2. On the other hand, for an approximate function value oracle $\sigma_0 > 0$, the theorem guarantees linear convergence for ratio feedback with a batchsize $\tilde{\mathcal{O}}(\kappa_\star^2)$, and for hypergradient feedback with a batchsize $\tilde{\mathcal{O}}(\kappa^2)$. These results are summarized in Table 1.

Large batchsizes are increasingly practical given modern computational constraints. It is typical to select a batchsize based on the number of processors available, as the time required is the same as needed for a smaller batchsize. Hence modern hardware accelerators like GPUs, with hundreds or thousands of parallel processors, reward algorithms that can make efficient use of large batchsizes.

We note that access to an exact function value oracle improves several aspects of the convergence theory. First, it reduces the batchsize requirement from $\tilde{\mathcal{O}}(\kappa^2)$ to $\tilde{\mathcal{O}}(1)$, as 1) with exact function values, the ratio feedback can be evaluated exactly, and 2) the stochastic online gradient $\frac{\nabla f_\zeta(x - P\nabla f_\xi(x))\nabla f(x)^\top}{f(x)-f^\star}$ is an unbiased gradient estimator of the deterministic ratio feedback $\frac{f(x-P\nabla f_\xi(x))-f^\star}{f(x)-f^\star}$. Moreover, an exact function value oracle improves the regret bound for ratio feedback from linear to sublinear, as shown by Lemma B.4. Finally, for the hypergradient feedback, an exact function value oracle makes it possible to ensure descent and prevent divergence.

We can combine these considerations to understand when OSGM-SGD offers an improved sample complexity compared to SGD. With an exact function value oracle, the sample complexity of OSGM-SGD with ratio feedback is $\tilde{\mathcal{O}}((1+\sigma_1^2)\kappa_\star \log \frac{1}{\epsilon})$, which improves the result of SGD from κ to a much smaller number $\kappa_\star$. The sample complexity of hypergradient feedback matches the result of SGD. Without an exact function value oracle, the sample complexity of ratio feedback is $\tilde{\mathcal{O}}((1+\sigma_1^2)\kappa_\star^3 \log \frac{1}{\epsilon})$, which improves on SGD when $\kappa_\star^3 < \kappa$. The sample complexity of hypergradient feedback is $\tilde{\mathcal{O}}((1+\sigma_1^2)\kappa^3 \log \frac{1}{\epsilon})$, which is worse than the result of SGD. However, our empirical experiments reveal that OSGM-SGD outperforms SGD and its variants with the same batchsize.

4 SOSGM with variance reduction

This section applies OSGM at the outer-loop level of SVRG to learn and adapt the matrix stepsize across epochs. Because the full gradient is available at each outer iterate, the feedback for stepsize selection is deterministic, and convergence analysis does not require the noise oracle assumed in Section 3.

We begin by recalling the SVRG method and a practically effective extension that uses heavy-ball momentum (Algorithm 3 with OSGM learning rate $\eta = 0$ and constant momentum $\beta_k = \beta$). SVRG uses a double-loop structure: at the start of outer epoch k, it computes the full gradient at a snapshot point $\tilde{x}^k$, and then performs m inner iterations using VR estimators. We also allow a heavy-ball momentum term parameterized by β inside the inner loop. When $\beta = 0$, the algorithm is SVRG. For $\beta > 0$, we will call the algorithm SVRG-HBM.

SVRG with acceleration has appeared in the literature [1, 36, 44, 49], but not with heavy-ball momentum. Experimentally, Nesterov acceleration and heavy-ball momentum perform about equally well, but heavy-ball momentum allows for theoretical guarantees in the context of OSGM that are currently unknown for Nesterov momentum [15].

4.1 OSGM-SVRG

In this section, we develop a unified framework OSGM-SVRG that uses OSGM at each outer iteration of SVRG to learn and improve the SVRG stepsize. The OSGM-SVRG framework is presented in Algorithm 3. Concretely, at outer iteration k, given the deterministic gradient $\nabla f(\tilde{x}^k)$ at the snapshot iterate, OSGM-SVRG uses either the ratio or hypergradient feedback with regularization $\rho > 0$,

$$r^{\rho}_{\tilde{x}^k}(P) = r_{\tilde{x}^k}(P) + \tfrac{\rho}{2}\|P\|_F^2 \quad \text{and} \quad h^{\rho}_{\tilde{x}^k}(P) = h_{\tilde{x}^k}(P) + \tfrac{\rho}{2}\|P\|_F^2, \tag{7}$$

where $r_{\tilde{x}^k}$ and $h_{\tilde{x}^k}$ are defined in (1). The regularization delivers an implicit bound on the size of the stepsize (Proposition 4.6) that will be important for the theoretical guarantees that follow. With this feedback design, apply online gradient descent to update the stepsize P_k,

$$P_k = \Pi_{\mathcal{P}}[(1-\eta\rho)P_{k-1} + \eta\nabla\ell_{\tilde{x}^k}(P_{k-1})], \quad \text{where} \quad \ell_{\tilde{x}^k} \in \{r_{\tilde{x}^k}, h_{\tilde{x}^k}\}. \tag{8}$$

In the inner-loop of SVRG (Line 9 of Algorithm 3), OSGM-SVRG moderates the learned stepsize with a decay factor $c \leq 1$ to ensure convergence in the context of stochastic gradient updates. We discuss the choice of decay factor further in Theorem 4.8.

Algorithm 3 OSGM-SVRG

1: **Input:** Initial iterate $\tilde{x}^0$, and stepsize P_0, epoch length m, OSGM learning rate η, decay factor c, candidate stepsize set $\mathcal{P}$, momentum sequence $\{\beta_k\}$, regularization ρ, and deterministic feedback function $\ell^{\rho}_{\tilde{x}^k} \in \{r^{\rho}_{\tilde{x}^k}, h^{\rho}_{\tilde{x}^k}\}$ defined by (7).
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute snapshot gradient $\nabla f(\tilde{x}^k) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\tilde{x}^k)$
4: Set $x^0 = \tilde{x}^k$
5: **for** $t = 0, \ldots, m-1$ **do**
6: Sample ξ^t uniformly
7: $g_t = \nabla f_{\xi^t}(x^t) - \nabla f_{\xi^t}(\tilde{x}^k) + \nabla f(\tilde{x}^k)$
8: $x^{t+1} = x^t - cP_k g_t + \beta_k(x^t - x^{t-1})$
9: **end for**
10: $P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta\nabla\ell^{\rho}_{\tilde{x}^k}(P_k)]$.
11: Choose $\tilde{x}^{k+1}$ uniformly from $\{x^0, \ldots, x^m\}$
12: **end for**

Remark 4.1. OSGM-SVRG is a general framework and can be reduced to SVRG and SVRG-HBM by setting learning rate $\eta = 0$, constant momentum $\beta_k = \beta$. The choice of regularization ρ is justified by Proposition 4.6 for scalar and matrix stepsize.

4.2 Convergence analysis

In this section, we establish a generic convergence result for OSGM-SVRG. First, we bound the potential decrease on expectation in each inner loop update. Within epoch k, define the filtration $\mathcal{F}_t$, the σ-algebra generated by all randomness up to and including step t. We write $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot \,|\, \mathcal{F}_t]$ for the conditional expectation with respect to this filtration.

Lemma 4.2 (Potential reduction)**.** For any epoch k and any inner step t, conditional on the filtration $\mathcal{F}_t$, the expected potential satisfies the following bounds:

(i) *No momentum.* Suppose $\beta_k = 0$ and assume stepsize $\underline{\alpha} I \preceq P_k \preceq \bar{\alpha} I$. Define $V^t := f(x^t) - f^\star$. Then

$$\mathbb{E}_t[V^{t+1}] \leq V^t - \left(2c\underline{\alpha}\mu - 2c^2\bar{\alpha}^2L^2\right)(f(x^t) - f^\star) + 2c^2\bar{\alpha}^2L^2(f(\tilde{x}^k) - f^\star). \tag{9}$$

For a scalar stepsize $P_k = \alpha I$, the upper and lower bounds $\underline{\alpha} = \bar{\alpha} = \alpha$ match.

(ii) *Bounded momentum.* Suppose $0 < \beta_k \leq \bar{\beta}$ and assume a scalar stepsize $P_k = \alpha I$. Define $V^t := f(x^t) - f^\star + \frac{L}{2}\|x^t - x^{t-1}\|^2$. Then

$$\begin{aligned}\mathbb{E}_t[V^{t+1}] \leq V^t &- \left(\tfrac{1}{2} - \bar{\beta}^2\right)L\|x^t - x^{t-1}\|^2 - (c\alpha - 2c^2\alpha^2L\kappa)\|\nabla f(x^t)\|^2 \\ &+ (1 - 2c\alpha L)\bar{\beta}\langle\nabla f(x^t), x^t - x^{t-1}\rangle + 4c^2\alpha^2L^2(f(\tilde{x}^k) - f^\star).\end{aligned} \tag{10}$$

Remark 4.3. The potential function defined in the bounded momentum setting (ii) is inspired by [35], which introduces potential function $f(x^t) - f^\star + \frac{1-\alpha L}{2\alpha}\|x^t - x^{t-1}\|^2$ to analyze the convergence of deterministic `HBM`. Given the bound on the potential reduction (10), we can choose the stepsize α and the upper bound of momentum $\bar{\beta}$ appropriately to guarantee a strict decrease of the potential.

From Lemma 4.2, we can directly derive the convergence of `SVRG` and `SVRG-HBM`.

Proposition 4.4 (Convergence of `SVRG` and `SVRG-HBM`)**.** Let decay factor $c = 1$. Use constant scalar stepsize $P_k = \alpha I$ and constant momentum $\beta_k = \beta$ in Algorithm 3.

(i) *`SVRG`.* Suppose momentum $\beta = 0$, and stepsize α satisfies $\alpha < \frac{1}{\kappa L}$. Then

$$\mathbb{E}[f(\tilde{x}^k) - f^\star] \leq \left(\tfrac{1+2m\alpha^2L^2}{2m(\alpha\mu - \alpha^2L^2)}\right)^k (f(\tilde{x}^0) - f^\star).$$

(ii) *`SVRG-HBM`.* Suppose momentum $\bar{\beta} \leq \sqrt{\alpha L - 2\alpha^2L^2\kappa}$, and stepsize $\alpha < \frac{1}{2\kappa L}$. Then

$$\mathbb{E}[f(\tilde{x}^k) - f^\star] \leq \left(\tfrac{1+4m\alpha^2L^2}{m(2\alpha\mu - 4\alpha^2L^2 - 2\bar{\beta}^2/\kappa)}\right)^k (f(\tilde{x}^0) - f^\star).$$

If $\alpha = \frac{1}{8\kappa L}$ and $\bar{\beta} = \frac{1}{\sqrt{32\kappa}}$, the contraction ratio is $\frac{8\kappa^2}{m} + \frac{1}{2}$.

Remark 4.5. From Proposition 4.4, to derive linear convergence, the epoch length of `SVRG` is $m = \mathcal{O}(\kappa^2)$, and the epoch length for `SVRG-HBM` is $m = \mathcal{O}(\kappa^2)$. Our result has a inferior dependence on κ compared to the result of [32], which requires $m = \mathcal{O}(\kappa)$. This difference results from the choice of potential function: [32] uses the potential $\|x^t - x^\star\|^2$. However, this potential cannot be directly extended to the matrix stepsize case where $\underline{\alpha} I \preceq P_k \preceq \bar{\alpha} I$. Therefore, in the analysis in this section, we use function value gap as the potential.

We now develop a convergence analysis for `OSGM-SVRG`. First, we show that the stepsize P_k is bounded. Then we show `OSGM-SVRG` converges linearly as long as the stepsize P is bounded, which we can ensure using stepsize decay ($c < 1$ in Algorithm 3) or projection (bounded $\mathcal{P}$) as a safeguard. This analysis approach is also used in the analysis of the stochastic `L-BFGS` method [43] and Barzilai-Borwein stepsize [54] in the `SVRG` framework.

Proposition 4.6 (Bounded stepsize)**.** Consider the stepsize P_k updated by `OSGM-SVRG`.

(i) *Scalar Stepsize.* Suppose $\mathcal{P} = \{\alpha I : \alpha \in \mathbb{R}\}$. Let $P_k = \alpha_k I$, regularization $\rho = 0$. Assume initial stepsize $\alpha_0 \in \left[\frac{1}{L}, \frac{1}{\mu}\right]$, and `OSGM` learning rate satisfies $\eta \leq \frac{1}{L}$ for hypergradient feedback, $\eta \leq \frac{1}{2L^2}$ for ratio feedback. Then

$$\alpha_k \in [\tfrac{1}{L}, \tfrac{1}{\mu}], \quad \forall k.$$

(ii) *Matrix Stepsize.* Suppose $\mathcal{P} = \mathbb{S}^n$ or $\mathcal{P} = \{\text{diag}(d) : d \in \mathbb{R}^n\}$. Let regularization $\rho = 2L$, learning rate $\eta \leq \frac{1}{L}$ for hypergradient feedback, $\rho = 4L^2$, $\eta \leq \frac{1}{2L^2}$ for ratio feedback. Assume initial stepsize $P_0 = 0$. Then

$$P_k \preceq \tfrac{1}{L} I, \quad \forall k.$$

Remark 4.7. Proposition 4.6 highlights the self-adaptivity of OSGM-SVRG. For scalar stepsize, even *without* regularization, the update rule automatically constrains the stepsize within a bounded interval. For matrix stepsizes, regularization is needed to keep $\|P_k\|_2$ upper-bounded.

With Lemma 4.2 and Proposition 4.6, we are ready to get the main convergence result.

Theorem 4.8 (Convergence of OSGM-SVRG)**.** Consider the following instantiations,

(i) *No momentum, scalar stepsize.* Suppose $\beta_k = 0$, $\mathcal{P} = \{\alpha I : \alpha \in \mathbb{R}\}$, and $c \leq \frac{1}{\kappa(\kappa+1)}$. Under the conditions in Proposition 4.6 (i), then

$$\mathbb{E}[f(\tilde{x}^k) - f^\star] \leq \left(\tfrac{\kappa}{2m(c-c^2\kappa)} + \tfrac{c\kappa^2}{1-c\kappa^2}\right)^k [f(\tilde{x}^0) - f^\star].$$

(ii) *No momentum, matrix stepsize.* Suppose $\beta_k = 0$, $\mathcal{P} = \{P \in \mathbb{S}^n : P \succeq \underline{\alpha} I\}$ or $\mathcal{P} = \{P = \text{diag}(d) : d \in \mathbb{R}^n, d \geq \underline{\alpha}\}$, and $c \leq \underline{\alpha}\mu$. Under the conditions in Proposition 4.6 (ii), then

$$\mathbb{E}[f(\tilde{x}^k) - f^\star] \leq \left(\tfrac{1}{2m(c\underline{\alpha}\mu-c^2)} + \tfrac{c}{(\underline{\alpha}\mu-c)}\right)^k [f(\tilde{x}^0) - f^\star].$$

(iii) *Bounded momentum, scalar stepsize.* Suppose $0 \leq \beta_k \leq \frac{1}{\kappa^2}$, $\mathcal{P} = \{\alpha I : \alpha \in \mathbb{R}\}$, $c \leq \frac{1}{4\kappa(\kappa+1)}$, and $\beta_k \leq \sqrt{\frac{c}{2}}$. Under the conditions in Proposition 4.6 (i), then

$$\mathbb{E}[f(\tilde{x}^k) - f^\star] \leq \left(\tfrac{\kappa}{m(c-c^2\kappa)} + \tfrac{4c\kappa^2}{1-4c\kappa^2}\right)^k [f(\tilde{x}^0) - f^\star].$$

Remark 4.9. Using Theorem 4.8, we can choose decay factor c and epoch length m to guarantee linear convergence with contraction rate $\frac{3}{4}$. For setting (i), a feasible choice is $c = \frac{1}{3\kappa^2}$ and $m = 9\kappa^3$. For setting (ii), a feasible choice is $c = \frac{\underline{\alpha}\mu}{3}$ and $m = \frac{9}{\underline{\alpha}^2\mu^2}$. For setting (iii), a feasible choice is $c = \frac{1}{12\kappa^2}$ and $m = 72\kappa^3$. This result has inferior dependence on κ compared to the results of SVRG and SVRG-HBM (Proposition 4.4). However, our method rewards practical faster convergence.

5 Experiments

The previous sections introduce OSGM-SGD and OSGM-SVRG. This section benchmarks the performance of these algorithms and some more practical variants on machine learning and deep learning tasks. Since $f^\star$ is typically unknown, all experiments use hypergradient feedback.

5.1 Practical variants

The algorithms OSGM-SGD and OSGM-SVRG are designed to admit clean convergence proofs. This section introduces variants of these algorithms optimized for performance rather than theoretical guarantees, which is summarized in Table 2. As shown in Appendix A.1, the practical variants can fail to converge in adversarial settings not covered by our theory. Nevertheless, they uniformly outperform the theoretical variants on the benchmarks we consider, suggesting that the counterexample conditions are rarely encountered in practice. As a result, numerical results in the main paper show only results for these practical variants. Results for the original, provably convergent variants, appear in Appendix D.3.1.

Family	Algorithm	Iteration	Feedback
SGD Variants	OSGM-SGD (Algorithm 2)	$x^{k+1} = x^k - P_k \nabla f_{\xi^k}(x^k)$	$\frac{f_{\xi^k}(x^{k+1}(P_k)) - f_{\xi^k}(x^k)}{\|\nabla f_{\xi^k}(x^k)\|^2}$
	OSGM-SGD (Algorithm 5)	$x^{k+1} = x^k - P_k \nabla f_{\xi^k}(x^k)$	$\frac{f_{\xi^k}(x^{k+1}(P_k)) - f_{\xi^k}(x^k)}{\|\nabla f_{\xi^k}(x^k)\|^2}$
Variance Reduction	OSGM-SVRG (Algorithm 3)	$x^{t+1} = x^t - c\, P_k\, g_t + \beta_t\, m^t$	$\frac{f(\tilde{x}^k - P_k \nabla f(\tilde{x}^k)) - f(\tilde{x}^k)}{\|\nabla f(\tilde{x}^k)\|^2} + \frac{\rho}{2}\|P_k\|_F^2$
	OSGM-SVRG(Algorithm 6)	$x^{t+1} = x^t - P_t\, g_t + \beta_t\, m^t$	$\frac{f_{\xi^t}(x^{t+1}(P_t,\beta_t)) - f_{\xi^t}(x^t)}{\|g_t\|^2 + \|m^t\|^2}$
	OSGM-SketchySVRG(Algorithm 4)	$x^{t+1} = x^t - \alpha_t \bar{P}_k\, g_t - D_t g_t + \beta_t\, m^t$	$\frac{f_{\xi^t}(x^{t+1}(\alpha_t,D_t,\beta_t)) - f_{\xi^t}(x^t)}{\|g_t\|^2 + \|m^t\|^2}$

Table 2: Summary of SOSGM algorithms. The parameters tuned by OSGM are marked as red. Methods with provable convergence guarantees are marked in colors. For OSGM-SketchySVRG, $\bar{P}_k$ is the random low-rank preconditioner updated by sketchy methods per epoch. We use $x^{k+1}(P_k)$ to denote parameterized update rule. For VR methods, $g_t := \nabla f_{\xi^t}(x^k) - \nabla f_{\xi^t}(\tilde{x}^k) + \nabla f(\tilde{x}^k)$ and $m^t := x^t - x^{t-1}$.

Practical OSGM-SGD variant and choice of feedback In Section 3, we discussed the choice of feedback function from a theoretical perspective. We saw that computing the gradient and evaluating the feedback on a different sample from the data distribution was necessary to guarantee convergence. However, in practice, this feedback can result in a conservative stepsize choice and slow convergence. Conversely, although feedback (3), which uses the same sample to compute the gradient and evaluate the feedback, may not converge in the worst case, it converges quickly in practice. Therefore, our numerical results in this section use variant OSGM-SGD powered by feedback (3). An empirical comparison between feedbacks (3) and (4) appears in Appendix D.3.2.

Practical OSGM-SVRG variants OSGM-SVRG applies OSGM as the outer-loop stepsize scheduler. In practice, we can apply OSGM in each inner loop and tune the stepsize and momentum simultaneously (as discussed in [15]). We use a diagonal stepsize since it is more efficient in memory and compute than a matrix stepsize. The pseudocode for the practical variant OSGM-SVRG appears as Algorithm 6 in Appendix D.2.

OSGM can be implemented on top of PROMISE methods [22], to tune the stepsize of a low rank preconditioner, or in the context of an optimizer with heavy-ball momentum, to tune the momentum coefficient [16]. Our strongest algorithm in practice, OSGM-SketchySVRG, uses OSGM to tune both the diagonal stepsize and momentum parameter of a heavy-ball variant of SketchySVRG. Pseudocode is presented as Algorithm 4.

VR methods offer linear convergence and perform best for statistical learning applications. In contrast, for non-convex problems such as training deep neural networks, VR methods tend to underperform. Hence our experiments showcase the methods on statistical learning, and SGD variants for deep learning.

5.2 Statistical learning

We benchmark OSGM-SVRG and OSGM-SketchySVRG on logistic regression with L2 regularization and ridge regression problems. We use datasets from LIBSVM [13] and OpenML [55], and set the batchsize to 256. The regularization parameter of logistic and ridge regression is $10^{-2}/n$. We present more details on the datasets in Appendix D.1.

Benchmark algorithms We benchmark the following variance reduction algorithms.

- *Baseline VR optimizers:* SVRG [32], SAGA [18], and L-Katyusha (Loopless Katyusha [34]) with tuned stepsize. For SVRG, the update frequency is $m = \lceil n/256 \rceil$.
- *PROMISE suite* [22]. SketchySVRG, SketchySAGA, and SketchyKatyusha with Nyström Subsampled Newton

Algorithm 4 `OSGM-SketchySVRG`

Input: Initial $\tilde{x}^0$, $P_0 = 0$, $\beta_0 = 0$, epoch length m, `OSGM` learning rates η_P and η_β, decay factor c, regularization ρ, candidate set of diagonal stepsize and momentum $\mathcal{P}$, $\mathcal{B}$
for $k = 0, 1, 2, \ldots$ **do**
  Compute snapshot gradient $\nabla f(\tilde{x}^k) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\tilde{x}^k)$
  Estimate low-rank preconditioner $\bar{P}_k$ by sketchy methods
  Set $x^0 = \tilde{x}^k$
  for $t = 0, \ldots, m-1$ **do**
    Sample ξ^t uniformly
    $g_t = \nabla f_{\xi^t}(x^t) - \nabla f_{\xi^t}(\tilde{x}^k) + \nabla f(\tilde{x}^k)$
    $x^{t+1} = x^t - \alpha_t \bar{P}_k g_t - D_t g_t + \beta_t (x^t - x^{t-1})$
    Define feedback $\ell_t(\alpha, D, \beta) = \frac{f(x^t - \alpha \bar{P}_k g_t - D g_t + \beta(x^t - x^{t-1})) - f(x^t)}{\|g_t\|^2 + \|x^t - x^{t-1}\|^2}$
    Update scalar stepsize: $\alpha_{t+1} = \alpha_t - \eta_P \nabla_\alpha \ell_t(\alpha_t, D_t, \beta_t)$
    Update diagonal stepsize: $D_{t+1} = \Pi_{\mathcal{P}}\big[D_t - \eta_P \nabla_D \ell_t(\alpha_t, D_t, \beta_t)\big]$
    Update momentum: $\beta_{t+1} = \Pi_{\mathcal{B}}\big[\beta_t - \eta_\beta \nabla_\beta \ell_t(\alpha_t, D_t, \beta_t)\big]$
  end for
  Choose $\tilde{x}^{k+1}$ uniformly from $\{x^0, \ldots, x^m\}$, set $\alpha_0 = \alpha_m, D_0 = D_m, \beta_0 = \beta_m$
end for

preconditioner, rank 10, and default stepsize.

- *Practical `OSGM-SVRG` variants.* `OSGM-SVRG` and `OSGM-SketchySVRG` with default `OSGM` learning rate for stepsize $\eta_P = 1/L$, and default learning rate for momentum $\eta_\beta = 0.1$.

We do not show performance of `SGD` or `OSGM-SGD` in our experiments because they perform much worse on benchmark tasks, as they converge sublinearly. We use default stepsize for `PROMISE` suite since it already significantly outperforms the tuned baseline optimizers as shown by the experiments in [22].

Suboptimality experiments Figure 1 shows performance plots for logistic regression and ridge regression. The y-axis represents the suboptimality $f(x) - f^\star$.

All these methods converge linearly. Yet a quick examination of the figures shows that as a practical matter, the baseline optimizers do *not* converge to a high-accuracy solution even after hundreds of epochs. In contrast, our strongest method `OSGM-SketchySVRG` reaches high-accuracy regimes (10^{-6} – 10^{-12}) normally considered beyond the reach of stochastic optimizers.

We observe that `OSGM-SVRG` always outperforms the baseline optimizers (`SVRG`, `SAGA`, `L-Katyusha`). `OSGM-SVRG` is competitive with `PROMISE` suite. However, `OSGM-SVRG` has lower memory and per-iteration compute cost, as methods of the `PROMISE` suite require storing a $d \times r$ matrix preconditioner (where r is the rank), while `OSGM-SVRG` stores only a d-dimensional vector to represent a diagonal preconditioner.

Following the discussion in [1], we emphasize the practical importance of high-accuracy solutions (e.g., function value gap $\leq 10^{-7}$). In particular, applications that use multiple black-box calls to ERM solvers [2, 24] can accumulate errors across calls, which makes high-accuracy solvers essential.

Performance experiments We compare the benchmark algorithms on a testbed of 47 medium-sized problems, including 31 logistic and 16 ridge regression problems. The primary metrics are the wall-clock time and the number of full data passes to reach suboptimality within 10^{-4} of the minimum. We set the budget as 600 seconds and 200 data passes.

Figure 2 shows the performance plots. On logistic regression, `OSGM-SketchySVRG` dominates the benchmark algorithms on both metrics and solves all the instances. `OSGM-SVRG` is comparable with the best `PROMISE` variant.

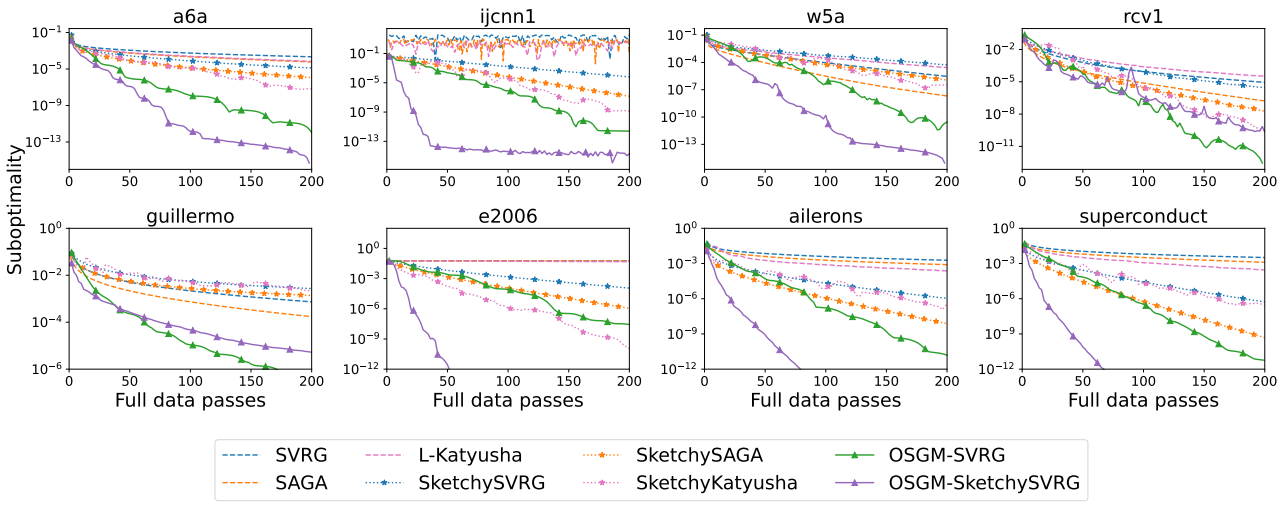

Figure 1: Suboptimality plots. First row: logistic regression. Second row: ridge regression.

On ridge regression, both `OSGM-SketchySVRG` and `OSGM-SVRG` tie for solving the most instances within the time budget. `OSGM-SketchySVRG` offers lower iteration counts, while `OSGM-SVRG` delivers the fastest solve times.

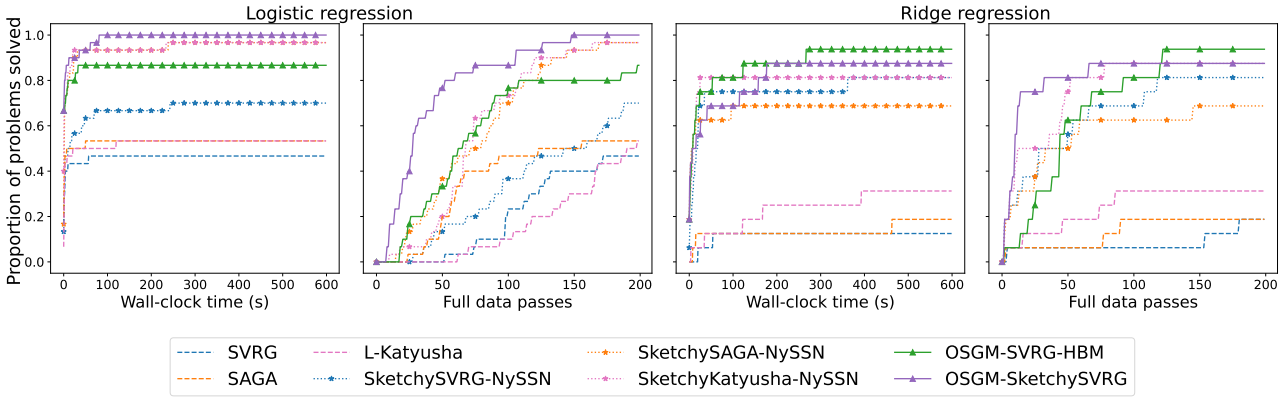

Figure 2: Performance plots. Left two: logistic regression. Right two: ridge regression.

5.3 Deep learning

We benchmark `OSGM-SGD` on training neural networks.

Benchmark problems We benchmark on the following problems, as in [6]: 1) Train an MLP model with two fully connected hidden layers on the MNIST dataset, and 2) train a VGG Net [51] on the CIFAR-10 image recognition dataset. For both benchmark problems, we use a batchsize 128 and weight decay 10^{-4}.

Benchmark algorithms We benchmark the following stochastic first-order algorithms:

- *Baseline optimizers.* `SGD`, `SGDN` (`SGD` with Nesterov momentum), and `Adam` [33] with stepsize 10^{-3} for both MLP and VGG tasks.

- *Hypergradient descent heuristics* [6]. `SGD-HD`, `SGDN-HD`, and `Adam-HD`. We use the parameter setting in [6]: the initial stepsize is 10^{-3} for both MLP and VGG tasks. `SGD-HD` and `SGDN-HD` use hypergradient learning rate 10^{-3}, and `Adam-HD` uses hypergradient learning rate 10^{-7} for MLP and 10^{-8} for VGG.
- `OSGM-SGD` with initial stepsize 10^{-3}, and `OSGM` learning rate 10^{-3} for MLP; 10^{-2} for VGG.

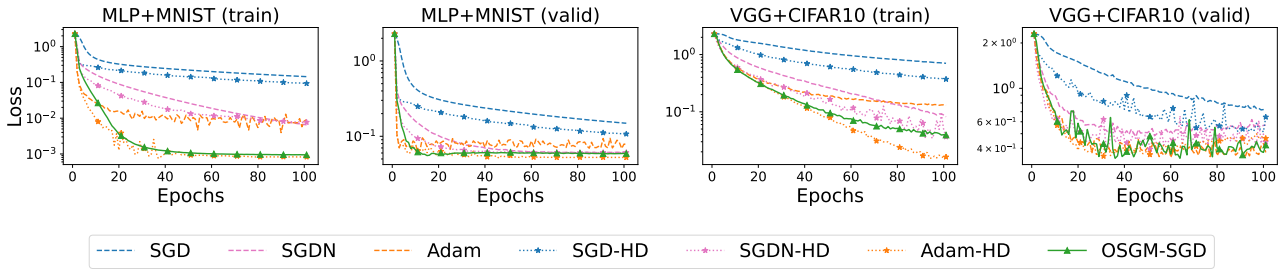

Figure 3: Performance of `OSGM-SGD`. Left two: training and validation loss of MLP on MNIST. Right two: training and validation loss of VGG on CIFAR10.

Performance plots Figure 3 shows the training and validation loss on MLP and VGG. `OSGM-SGD` significantly outperforms the baseline methods (`SGD`, `SGDN`, and `Adam`) on both training and validation sets. `OSGM-SGD` also outperforms `SGD-HD` uniformly, showing the advantage of feedback function in `OSGM`. Notably, `OSGM-SGD` is competitive with `Adam-HD`, demonstrating that simply using adaptive stepsize for `SGD` can match the performance of diagonal scaled momentum methods.

6 Conclusion

In this work, we introduce `SOSGM`, an extension of `OSGM` that learns a matrix stepsize for stochastic gradient methods. We propose the `OSGM-SGD` and `OSGM-SVRG` algorithms, prove linear convergence for `OSGM-SVRG` and high-probability convergence guarantees for `OSGM-SGD` in the large-batch regime. Numerical experiments show the advantage of these methods especially on ill-conditioned machine learning problems.

References

[1] Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, 18(221):1–51, 2018. 1, 9, 13, 35

[2] Zeyuan Allen-Zhu, Zhenyu Liao, and Yang Yuan. Optimization algorithms for faster computational geometry. *arXiv preprint arXiv:1412.1001*, 2014. 13

[3] Luís B Almeida, Thibault Langlois, José D Amaral, and Alexander Plakhov. Parameter adaptation in stochastic optimization. In *On-line learning in neural networks*, pages 111–134. 1999. 2

[4] Amit Attia and Tomer Koren. A note on high-probability analysis of algorithms with exponential, sub-gaussian, and general light tails. *arXiv preprint arXiv:2403.02873*, 2024. 22, 24

[5] Jean-François Aujol, Jérémie Bigot, and Camille Castera. Stochastic adaptive gradient descent without descent. *arXiv preprint arXiv:2509.14969*, 2025. 3

[6] Atilim Güneş Baydin, Robert Cornish, David Martínez Rubio, Mark Schmidt, and Frank Wood. Online learning rate adaptation with hypergradient descent. In *Sixth International Conference on Learning Representations (ICLR), Vancouver, Canada, April 30 – May 3, 2018*, 2018. 2, 14, 15

[7] Albert S Berahas, Liyuan Cao, and Katya Scheinberg. Global convergence rate analysis of a generic line search algorithm with noise. *SIAM Journal on Optimization*, 31(2):1489–1518, 2021. 7

[8] Albert S Berahas, Liyuan Cao, Krzysztof Choromanski, and Katya Scheinberg. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *Foundations of Computational Mathematics*, 22(2):507–560, 2022. 7

[9] Raghu Bollapragada, Richard H Byrd, and Jorge Nocedal. Exact and inexact subsampled newton methods for optimization. *IMA Journal of Numerical Analysis*, 39(2):545–578, 2019. 1, 2

[10] Richard H Byrd, Gillian M Chin, Jorge Nocedal, and Yuchen Wu. Sample size selection in optimization methods for machine learning. *Mathematical programming*, 134(1):127–155, 2012. 7

[11] Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016. 1, 2

[12] Richard G Carter. On the global convergence of trust region algorithms using inexact gradient information. *SIAM Journal on Numerical Analysis*, 28(1):251–265, 1991. 7

[13] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011. 12

[14] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36:49205–49233, 2023. 3

[15] Ya-Chi Chu, Wenzhi Gao, Yinyu Ye, and Madeleine Udell. Gradient methods with online scaling part ii. practical aspects. *arXiv preprint arXiv:2509.11007*, 2025. 1, 2, 4, 9, 12, 34

[16] Ya-Chi Chu, Wenzhi Gao, Yinyu Ye, and Madeleine Udell. Provable and practical online learning rate adaptation with hypergradient descent. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=NkVCB1Cpgl`. 2, 12

[17] Aaron Defazio and Konstantin Mishchenko. Learning-rate-free learning by d-adaptation. In *International Conference on Machine Learning*, pages 7449–7479. PMLR, 2023. 3

[18] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014. 1, 12, 35

[19] Aaron Defazio, Xingyu Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. *Advances in Neural Information Processing Systems*, 37:9974–10007, 2024. 3

[20] Michał Dereziński. Stochastic variance-reduced newton: Accelerating finite-sum minimization with large batches. *arXiv preprint arXiv:2206.02702*, 2022. 1, 2

[21] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011. 1, 3

[22] Zachary Frangella, Pratik Rathore, Shipu Zhao, and Madeleine Udell. Promise: Preconditioned stochastic optimization methods by incorporating scalable curvature estimates. *Journal of Machine Learning Research*, 25(346):1–57, 2024. 1, 3, 12, 13

[23] Zachary Frangella, Pratik Rathore, Shipu Zhao, and Madeleine Udell. Sketchysgd: reliable stochastic optimization via randomized curvature estimates. *SIAM Journal on Mathematics of Data Science*, 6(4): 1173–1204, 2024. 1, 3

[24] Roy Frostig, Cameron Musco, Christopher Musco, and Aaron Sidford. Principal component projection without principal component analysis. In *International Conference on Machine Learning*, pages 2349–2357. PMLR, 2016. 13

[25] Wenzhi Gao, Ya-Chi Chu, Yinyu Ye, and Madeleine Udell. Gradient methods with online scaling part i. theoretical foundations. *arXiv preprint arXiv:2505.23081*, 2025. 1, 2, 4

[26] Wenzhi Gao, Ya-Chi Chu, Yinyu Ye, and Madeleine Udell. Gradient methods with online scaling. In Nika Haghtalab and Ankur Moitra, editors, *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pages 2192–2226. PMLR, 30 Jun–04 Jul 2025. URL `https://proceedings.mlr.press/v291/gao25a.html`. 2

[27] Robert Gower, Donald Goldfarb, and Peter Richtárik. Stochastic block bfgs: Squeezing more curvature out of data. In *International Conference on Machine Learning*, pages 1869–1878. PMLR, 2016. 1, 2

[28] Robert M Gower, Mark Schmidt, Francis Bach, and Peter Richtárik. Variance-reduced methods for machine learning. *Proceedings of the IEEE*, 108(11):1968–1983, 2020. 21

[29] Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pages 1842–1850. PMLR, 2018. 3

[30] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012. 1

[31] Robert A Jacobs. Increased rates of convergence through learning rate adaptation. *Neural networks*, 1(4): 295–307, 1988. 2

[32] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013. 1, 10, 12, 35

[33] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1, 3, 14

[34] Dmitry Kovalev, Samuel Horváth, and Peter Richtárik. Don't jump through hoops and remove those loops: Svrg and katyusha are better without the outer loop. In *Algorithmic learning theory*, pages 451–467. PMLR, 2020. 3, 12, 35

[35] Anastasiya Kulakova, Marina Danilova, and Boris Polyak. Non-monotone behavior of the heavy ball method. *arXiv preprint arXiv:1811.00658*, 2018. 10

[36] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. *Advances in neural information processing systems*, 28, 2015. 9

[37] Hong Liu, Zhiyuan Li, David Leo Wright Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=3xHDeA8Noi`. 3

[38] Yanli Liu, Fei Feng, and Wotao Yin. Acceleration of svrg and katyusha x by inexact preconditioning. In *International Conference on Machine Learning*, pages 4003–4012. PMLR, 2019. 3

[39] Nicolas Loizou, Sharan Vaswani, Issam Hadj Laradji, and Simon Lacoste-Julien. Stochastic polyak step-size for sgd: An adaptive learning rate for fast convergence. In *International Conference on Artificial Intelligence and Statistics*, pages 1306–1314. PMLR, 2021. 3

[40] Ashique Rupam Mahmood, Richard S Sutton, Thomas Degris, and Patrick M Pilarski. Tuning-free step-size adaptation. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2121–2124. IEEE, 2012. 2

[41] Yura Malitsky and Konstantin Mishchenko. Adaptive gradient descent without descent. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020. 3

[42] James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, pages 2408–2417. PMLR, 2015. 3

[43] Philipp Moritz, Robert Nishihara, and Michael Jordan. A linearly-convergent stochastic l-bfgs algorithm. In *Artificial intelligence and statistics*, pages 249–258. PMLR, 2016. 1, 2, 10

[44] Atsushi Nitanda. Stochastic proximal gradient descent with acceleration techniques. *Advances in neural information processing systems*, 27, 2014. 9

[45] Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019. 4

[46] Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled newton methods. *Mathematical Programming*, 174(1):293–326, 2019. 1, 2

[47] David Martinez Rubio. Convergence analysis of an adaptive method of gradient descent. *University of Oxford, Oxford, M. Sc. thesis*, 2017. 2

[48] Nicol N Schraudolph. Local gain adaptation in stochastic gradient descent. 1999. 2

[49] Fanhua Shang, Yuanyuan Liu, James Cheng, and Jiacheng Zhuo. Fast stochastic variance reduced gradient method with momentum acceleration for machine learning. *arXiv preprint arXiv:1703.07948*, 2017. 9

[50] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR, 2018. 3

[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1409.1556`. 14

[52] Jingruo Sun, Zachary Frangella, and Madeleine Udell. Sapphire: Preconditioned stochastic variance reduction for faster large-scale statistical learning. *arXiv preprint arXiv:2501.15941*, 2025. 3

[53] Richard S Sutton. Adapting bias by gradient descent: An incremental version of delta-bar-delta. In *AAAI*, volume 92, pages 171–176. Citeseer, 1992. 2

[54] Conghui Tan, Shiqian Ma, Yu-Hong Dai, and Yuqiu Qian. Barzilai-borwein stepsize for stochastic gradient descent. *Advances in neural information processing systems*, 29, 2016. 10

[55] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. 12

[56] Sharan Vaswani and Reza Babanezhad Harikandeh. Armijo line-search can make (stochastic) gradient descent provably faster. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=LKQIS65fgd`. 3

[57] Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. *Advances in neural information processing systems*, 32, 2019. 3

[58] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018. 21, 22

[59] Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M. Kakade. SOAP: Improving and stabilizing shampoo using adam for language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=IDxZhXrpNf`. 3

[60] Zhuang Yang. Adaptive powerball stochastic conjugate gradient for large-scale learning. *IEEE Transactions on Big Data*, 9(6):1598–1606, 2023. doi: 10.1109/TBDATA.2023.3300546. 2

[61] Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. Adahessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10665–10673, 2021. 3

Appendix

Table of Contents

A Details of counterexample

Example A.1. Consider an instantiation of problem (2) with $n = 2$:

$$f_1(x) = \begin{cases} 0, & x \le 0 \\ \frac{1}{2}x^2 & 0 < x \le 1 \\ x - 0.5, & x > 1 \end{cases}, \quad f_2(x) = \begin{cases} -x - 0.5, & x \le -1 \\ \frac{1}{2}x^2, & -1 < x \le 0 \\ 0, & x > 0 \end{cases}. \tag{11}$$

The minimizer of f is $x^\star = 0$ with $f^\star = 0$, and $x^\star \in \arg\min_x f_1(x) \cap \arg\min_x f_2(x)$. This problem satisfies the interpolation condition, and each f_i is convex and 1-smooth.

We show that if `OSGM` is initialized at $x^0 = 1$, $P^0 = 2$, the `OSGM` iteration using the naive feedback Eq. (3) does not converge: instead, $\|x^k - x^\star\| = 1$ for every iterate x^k.
Proof: Start from (x^0, P^0). *Case 1*: $\xi^0 = 1$, then $\nabla f_1(x^0 - P_0 \nabla f_1(x^0)) = 0$. From the `OSGM` update with feedback (3), P is not updated and we have $(x^1, P^1) = (-1, 2)$. *Case 2*: $\xi^0 = 2$, then $\nabla f_2(x^0) = 0$ and $(x^1, P^1) = (1, 2)$. In either case, the stepsize P is not updated and the iterate x does not contract to the optimal solution $x^\star$.

We can generalize the proof of Example A.1 to show that even if $P^0 = 0$ and the `OSGM` learning rate is arbitrary $\eta > 0$, there exists $x^0 > 1$ such that `OSGM` with hypergradient feedback $h_{x^k,\xi^k}(P_k)$ (3) fails to converge. Since $\nabla f_2(x) = 0$ for all $x \ge 0$, whenever $\xi^k = 2$ for $x^k \ge 0$, both stepsize P_k and iterate x^k are unchanged. So for simplicity, we will number only the iterates where $\xi^k = 1$. From the `OSGM` update, we obtain $P_k = \eta k$, and $x^k = x^0 - \frac{(k-1)k}{2}\eta$. If $t = \lceil \frac{2}{\eta} \rceil$ and $x^0 = \frac{(t-1)t}{2}\eta + 1$, then $x^t = 1, P_t \ge 2$. Then repeating the argument in

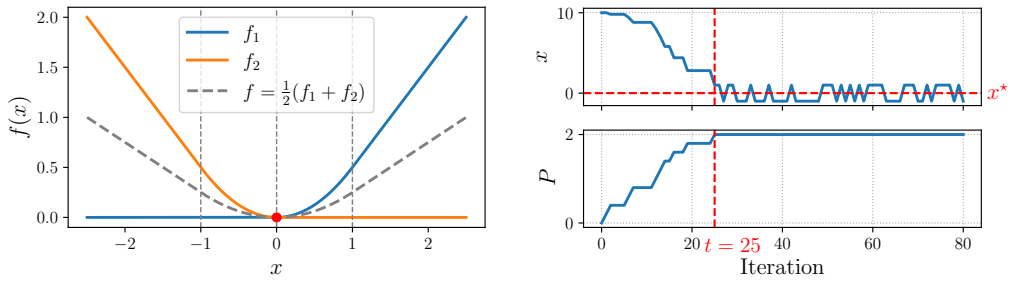

Figure 4: Illustration of Example A.1. Left: plot of f defined by (11). Right: behaviour of SOSGM with naive hypergradient feedback (3), with $x_0 = 10$, $P_0 = 0$, and $\eta = 0.1$.

the previous paragraph concludes that the algorithm does not converge. This behaviour is also visualized in Figure 4. After iteration $t = 25$, stepsize P stabilizes at 2, and iterate x oscillates between 1 and -1.

It is worth noting that the above failure mode of (3) also applies when OSGM is used to tune the stepsize of VR methods. In particular, consider the ideal (but typically unavailable) VR estimator $g^k := \nabla f_{\xi^k}(x^k) - \nabla f_{\xi^k}(x^\star)$ [28]. In Example A.1, we have $\nabla f_{\xi^k}(x^\star) = 0$ for every sample ξ^k, and hence $g^k = \nabla f_{\xi^k}(x^k)$; the resulting dynamics coincide with the stochastic gradient case above. Therefore, applying OSGM with the naive in-sample feedback (3) to update the stepsize within the inner (stochastic) iterations of a VR method can also fail to converge. This does not conflict with OSGM-SVRG, which updates the stepsize only in the outer loop using deterministic feedback based on the full gradient at the snapshot point.

The issue in the counterexample is that the learned stepsize and resulting update can be (locally) optimal for each sampled objective f_{ξ^k}, while still being suboptimal for f. This suboptimality with respect to f is not captured by the naive feedbacks (3), which evaluate progress only on the selected mini-batch. We therefore require feedbacks that better reflect progress on the full objective f.

B Proofs of results in Section 3

B.1 Proof of Example 3.2

Proof. Let $H := \frac{1}{n}\sum_{i=1}^n a_i a_i^\top$, the deviation of the stochastic gradient is

$$\nabla f_i(x) - \nabla f(x) = (a_i a_i^\top - H)(x - x^\star) = \Delta_i(x - x^\star).$$

Since the feature vectors a_i are sub-Gaussian with parameter σ_a, meaning $\langle a_i, u\rangle$ is sub-Gaussian with parameter at most $\sigma_a\|u\|$ for any $u \in \mathbb{R}^d$. Then standard results on sub-Gaussian quadratic forms [58] imply the deviation bound

$$\mathbb{P}\{\|\Delta_i v\| \geq t\|v\|\} \leq 2\exp\big(-\tfrac{c\,t^2}{\sigma_a^4}\big) \qquad \forall v \neq 0,$$

for an absolute constant $c > 0$. Applying this with $v = x - x^\star$ and $\|\nabla f(x)\| \geq \lambda_{\min}(H)\,\|x - x^\star\|$, we obtain

$$\mathbb{P}\{\|\nabla f_i(x) - \nabla f(x)\| \;\geq\; t\,\|\nabla f(x)\|\} \leq 2\exp\big(-\tfrac{c\,(t\lambda_{\min}(H))^2}{\sigma_a^4}\big),$$

which is precisely Assumption 3.1 (iii) with $\sigma_1 = \Theta(\sigma_a^2/\lambda_{\min}(H))$.

Moreover, the same model also satisfies the function-value oracle Assumption 3.2. Writing $u := x - x^\star$, we have

$$f_i(x) = \tfrac{1}{2}(a_i^\top u)^2, \qquad f(x) = \tfrac{1}{2}u^\top H u,$$

so

$$f_i(x) - f(x) = \tfrac{1}{2}\big[(a_i^\top u)^2 - u^\top H u\big].$$

Since $a_i^\top u$ is sub-Gaussian with parameter at most $\sigma_a\|u\|$, the Hanson–Wright inequality [58] implies that

$$\mathbb{P}\big\{\big|(a_i^\top u)^2 - u^\top H u\big| \geq 2t\, u^\top H u\big\} \leq 2\exp\big(-\tfrac{\tilde{c}\,(t\lambda_{\min}(H))^2}{\sigma_a^4}\big),$$

for some absolute constant $\tilde{c} > 0$. Using $|f_i(x) - f(x)| = \frac{1}{2}|(a_i^\top u)^2 - u^\top H u|$ and $|f(x) - f(x^\star)| = \frac{1}{2}u^\top H u$, this yields

$$\mathbb{P}\{|f_i(x) - f(x)| \;\geq\; t\,|f(x) - f(x^\star)|\} \leq 2\exp\big(-\tfrac{\tilde{c}\,(t\lambda_{\min}(H))^2}{\sigma_a^4}\big),$$

which matches Assumption 3.2 with $\sigma_0 = \Theta(\sigma_a^2/\lambda_{\min}(H))$. In particular, both the gradient oracle and the function-value oracle have relative noise levels of the same order, $\sigma_0 \asymp \sigma_1$. □

B.2 Proof sketch and lemmas

The proof sketch is as follows. We first establish the convergence guarantee for OSGM-SGD under the deterministically bounded oracles (Assumptions B.1 and B.2). Then, notice that Assumptions 3.1 and 3.2 implies Assumptions B.1 and B.2 uniformly over the iterates with high probability. Applying the general reduction framework introduced in [4], we obtain the high-probability convergence guarantee for OSGM-SGD under Assumptions 3.1 and 3.2 with only a small loss in logarithmic factors.

Assumption B.1. The stochastic gradient oracle $\nabla f_\xi(x)$ satisfies, $\mathbb{E}_\xi[\nabla f_\xi(x)] = \nabla f(x)$, and for any iterate $x \in \{x^k\}_{k=1,2,\ldots}$ and any $t > 0$,

$$\|\nabla f_\xi(x) - \nabla f(x)\| \leq \sigma_1\|\nabla f(x)\|.$$

Assumption B.2. The function value oracle $f_\xi(x)$ is unbiased, $\mathbb{E}_\xi[f_\xi(x)]{=}f(x)$, and for any iterate $x \in \{x^k\}_{k=1,2,\ldots}$ and any $t > 0$,

$$|f_\xi(x) - f(x)| \leq \sigma_0|f(x) - f(x^\star)|.$$

In the following subsections, we establish the convergence guarantee for OSGM-SGD under the deterministically bounded oracles Assumptions B.1 and B.2. In Section B.2.1, we introduce the proxy feedback $\hat{r}_{x,\xi}, \hat{h}_{x,\xi,\zeta}$, analyze its hindsight performance and establish its reduction to the convergence guarantee. In Section B.2.2, we establish the regret bounds of doing OGD on feedback $r_{x,\xi,\zeta}, h_{x,\xi,\zeta}$ with respect to the proxy feedback. In Section B.2.3, we derive convergence guarantee under deterministic oracles follows by combining these results together.

B.2.1 Feedback design

Define the proxy ratio and hypergradient loss as,

$$\hat{r}_{x,\xi}(P) = \tfrac{f(x - P\nabla f_\xi(x)) - f^\star}{f(x) - f^\star}, \quad \hat{h}_{x,\xi,\zeta}(P) = \tfrac{f_\zeta(x - P\nabla f_\xi(x)) - f_\zeta(x)}{\|\nabla f(x)\|^2}. \tag{12}$$

Notice that these feedbacks are only used for analysis purpose, not used in OSGM-SGD since their gradients are not available. We begin by analyzing the properties of feedback functions.

Lemma B.1 (Properties of feedback functions)**.** Under Assumption B.1, then for any iterate $x \in \{x^k\}_{k=1,2,\ldots}$ and for all ξ, the following statements hold.

(i) $\hat{r}_{x,\xi}$ is convex and $2L^2(1+\sigma_1)^2$-smooth.

(ii) Suppose $\sigma_1 < 1$, $r_{x,\xi,\zeta}$ is convex, $2L\left(LD + \frac{1+\sigma_1}{1-\sigma_1}\right)$-Lipschitz, and $2L^2$-smooth.

(iii) $\hat{h}_{x,\xi,\zeta}$ is convex and $L(1+\sigma_1)^2$-smooth.

(iv) Suppose $\sigma_1 < 1$, $h_{x,\xi,\zeta}$ is convex, $\left(LD + \frac{1+\sigma_1}{1-\sigma_1}\right)$-Lipschitz, and L-smooth.

Lemma B.2 (Hindsight feedback)**.** Under Assumption B.1, there exists hindsight stepsize $P_\star^r, P_\star^h$ such that

(i) Ratio feedback, strongly convex. Suppose $\mu > 0$, for any $\delta \in (0,1)$, w.p. $1 - \frac{\delta}{2}$,

$$\sum_{k=1}^K \hat{r}_{x^k,\xi^k}(P_\star^r) \le \left(1 - \frac{1}{\kappa_\star\left(1+\frac{\kappa_\star}{\kappa}\sigma_1^2\right)}\right)K + \frac{2\kappa_\star\sigma_1^3}{(\kappa+\kappa_\star\sigma_1^2)^2}\sqrt{2K\log\tfrac{2}{\delta}}. \tag{13}$$

where $\kappa_\star \le \kappa$ is the condition number by applying preconditioner P such that $\frac{1}{\kappa_\star}P^{-1} \preceq \nabla^2 f(x) \preceq P^{-1}$.

(ii) Hypergradient feedback, convex. Suppose $\mu \ge 0$, for any $\delta \in (0,1)$, w.p. $1-\delta$,

$$\sum_{k=1}^K \hat{h}_{x^k,\xi^k,\zeta^k}(P_\star^h) \le -\frac{K}{2L(1+\sigma_1^2)} + \left[\frac{\sigma_1^3}{L(1+\sigma_1^2)^2} + \frac{\sigma_1(1+\sigma_1)}{L(1+\sigma_1^2)}\right]\sqrt{2K\log\tfrac{1}{\delta}}. \tag{14}$$

Remark B.3. The hindsight convergence of ratio feedback (13) suggests that, when the underlying condition number κ is fixed, improving the preconditioner, or decreasing $\kappa_\star$, reduces the impact of stochastic gradient noise σ_1.

B.2.2 Regret analysis

Suppose $\{P_1, \dots, P_k\}$ is obtained from `OSGM-SGD`, we have the following regret guarantee for both ratio and hypergradient feedback.

Lemma B.4 (Regret bounds)**.** Under Assumptions B.1 and B.2. Then for any preconditioner $P_\star \in \mathcal{P}$,

(i) *Ratio feedback.* Suppose $\sigma_1 < 1$, and $f_\star = \min_x f_\xi(x)$ for all ξ, for any $\delta \in (0,1)$, w.p. $1 - \frac{\delta}{2}$,

$$\begin{aligned}\sum_{k=1}^K (\hat{r}_{x^k,\xi^k}(P_k) - \hat{r}_{x^k,\xi^k}(P_\star)) \le{}& 2LD\left(LD + \tfrac{1+\sigma_1}{1-\sigma_1}\right)\sqrt{K} + 2\sigma_0 LD\left(LD + \tfrac{1}{1-\sigma_1}\right)K \\ &+ 2\sigma_1 LD\left(LD + \tfrac{1}{1-\sigma_1}\right)\sqrt{2K\log\tfrac{2}{\delta}}.\end{aligned} \tag{15}$$

When $\sigma_0 = 0$, the function value is exact, the regret bound is sublinear,

$$\sum_{k=1}^K (\hat{r}_{x^k,\xi^k}(P_k) - \hat{r}_{x^k,\xi^k}(P_\star)) \le 2LD\left(LD + \tfrac{1+\sigma_1}{1-\sigma_1}\right)\sqrt{K} + 2\sigma_1 LD\left(LD + \tfrac{1}{1-\sigma_1}\right)\sqrt{2K\log\tfrac{2}{\delta}}.$$

(ii) *Hypergradient feedback.* Suppose $\sigma_1 < 1$,

$$\sum_{k=1}^K (\hat{h}_{x^k,\xi^k}(P_k) - \hat{h}_{x^k,\xi^k}(P_\star)) \le D\left(LD + \tfrac{1+\sigma_1}{1-\sigma_1}\right)\sqrt{K} + 3\sigma_1 D\left(LD + \tfrac{1+\sigma_1}{1-\sigma_1}\right)K. \tag{16}$$

B.2.3 Iteration complexity

Before establishing convergence under high-probability assumption, we first prove convergence under the deterministic assumption.

Proposition B.5 (Convergence with deterministically bounded oracles)**.** Under Assumptions B.1 and B.2. Running `SOSGM` for K iterations, then for any $\delta \in (0,1)$, w.p. $1-\delta$,

(i) *Ratio feedback, strongly convex.* Suppose $\mu > 0$, then for any $\sigma_1 < \frac{1}{2}$, $\sigma_0 = \mathcal{O}\left(\frac{1}{L^2D^2\kappa_\star}\right)$,

$$\frac{f(x^K)-f(x^\star)}{f(x^0)-f(x^\star)} \le \left(1 - \frac{1}{2\left(1+\frac{\kappa_\star}{\kappa}\sigma_1^2\right)\kappa_\star} + \mathcal{O}\left(\frac{1}{\sqrt{K}}\sqrt{\log\tfrac{1}{\delta}}\right)\right)^K.$$

(ii) *Hypergradient feedback, strongly convex.* Suppose $\mu > 0$, then for $\sigma_1 = \mathcal{O}\left(\frac{1}{L^2D^2}\right)$, $\sigma_0 = \mathcal{O}\left(\frac{1}{\kappa}\right)$,

$$\frac{f(x^K)-f(x^\star)}{f(x^0)-f(x^\star)} \le \left(1 - \frac{1}{2(1+\sigma_1^2)\kappa} + \mathcal{O}\left(\frac{1}{\sqrt{K}}\sqrt{\log\frac{1}{\delta}}\right)\right)^K.$$

(iii) *Hypergradient feedback, convex.* Suppose $\mu = 0$, then for $\sigma_1 = \mathcal{O}\left(\frac{1}{L^2D^2}\right)$, $\sigma_0 = 0$,

$$f(x^K) - f(x^\star) \le \min\left\{\frac{\Delta^2}{\max\left\{K\left(\frac{1}{4L(1+\sigma_1^2)} - \mathcal{O}\left(\frac{1}{\sqrt{K}}\sqrt{\log\frac{1}{\delta}}\right)\right), 0\right\}}, f(x^0) - f^\star\right\}.$$

The exact expression is shown in the proof.

B.3 Proof of Theorem 3.3

Now we prove Theorem 3.3 from Proposition B.5. We use the general reduction framework from bounded stochastic oracles to light-tailed oracles. We rephrase the result for completeness.

Lemma B.6 (Reduction framework from bounded to light-tailed oracles [4])**.** Given an algorithm $\mathcal{A}$, number of rounds K, and a sub-Gaussian sampling oracle $\mathcal{O}$, there exists a B-bounded sampling oracle $\widetilde{\mathcal{O}}$ with

$$B = 4\sigma\sqrt{\max\left\{\log\frac{4K}{\delta}, 1\right\}},$$

such that $\mathbb{E}[\mathcal{O}(x)] = \mathbb{E}[\widetilde{\mathcal{O}}(x)]$ for all queries $x \in \mathcal{X}$, and with probability at least $1-\delta$, the outputs of algorithm $\mathcal{A}$ with $\mathcal{O}$ and $\widetilde{\mathcal{O}}$ are identical.

Lemma B.6 implies that for analyzing an algorithm with light-tailed oracles, it suffices to analyze a simpler version of the algorithm that uses the bounded oracles. The results derived from bounded oracles equally apply to the original algorithm with only a small loss in logarithmic factors in K.

As a direct application of Lemma B.6, we get the convergence of `SOSGM` under sub-Gaussian Assumptions.

Proof of Theorem 3.3. By Lemma B.6, substituting $\sigma \leftarrow 4\sigma\sqrt{\max\left\{\log\frac{4K}{\delta}, 1\right\}}$ for both $\sigma = \sigma_0$ and $\sigma = \sigma_1$ into Proposition B.5 finishes the proof. □

B.4 Proof of lemmas

B.4.1 Proof of Lemma B.1

Proof. Notice that $f(x - P\nabla f_\xi(x))$ is convex in P since it is the composition between affine function $x - P\nabla f_\xi(x)$ and convex function f. Therefore $\hat{r}_{x,\xi}$ is convex because it simply translates and scales $f(x - P\nabla f_\xi(x))$ by a positive factor $f(x) - f^\star$. Similarly, $\hat{h}_{x,\xi}$ is convex.

Also, $f_\zeta(x - P\nabla f_\xi(x))$ is convex in P since it is the composition between affine function $x - P\nabla f_\xi(x)$ and convex function f_ζ. Therefore $r_{x,\xi,\zeta}$ and $h_{x,\xi,\zeta}$ are convex.

Then we consider the smoothness of the feedback functions. First we prove that $u_{x,\xi}(P) = f(x - P\nabla f_\xi(x))$ is $L\|\nabla f_\xi(x)\|^2$-smooth. For any P_1, P_2,

$$\begin{aligned}
\|\nabla u_{x,\xi}(P_1) - \nabla u_{x,\xi}(P_2)\|_F &= \|(\nabla f(x - P_1\nabla f_\xi(x)) - \nabla f(x - P_2\nabla f_\xi(x)))\nabla f_\xi(x)\|_F \\
&= \|\nabla f(x - P_1\nabla f_\xi(x)) - \nabla f(x - P_2\nabla f_\xi(x))\|\|\nabla f_\xi(x)\| \\
&\le L\|\nabla f_\xi(x)\|^2\|P_1 - P_2\|_F,
\end{aligned}$$

Similarly, let $\hat{u}_{x,\xi,\zeta}(P) = f_\zeta(x - P\nabla f_\xi(x))$, $\hat{u}_{x,\xi,\zeta}$ is $L\|\nabla f_\xi(x)\|^2$-smooth.

where the inequality is because f is L-smooth. Notice that

$$\hat{r}_{x,\xi}(P) = \tfrac{u_{x,\xi}(P)-f^\star}{f(x)-f^\star}, \hat{h}_{x,\xi,\zeta}(P) = \tfrac{\hat{u}_{x,\xi,\zeta}(P)-f(x)}{\|\nabla f(x)\|^2}, h_{x,\xi,\zeta}(P) = \tfrac{\hat{u}_{x,\xi,\zeta}(P)-f_\zeta(x)}{\|\nabla f_\xi(x)\|^2}, r_{x,\xi,\zeta}(P) = \tfrac{\hat{u}_{x,\xi,\zeta}(P)-f^\star}{f_\xi(x)-f^\star}$$

Apply the inequality $\|\nabla f_\xi(x)\| \le (1+\sigma_1)\|\nabla f(x)\|$ from Assumption B.1. The smoothness of $\hat{r}_{x,\xi}$ is $\frac{L\|\nabla f_\xi(x)\|^2}{f(x)-f^\star} \le (1+\sigma_1)^2\frac{L\|\nabla f(x)\|^2}{f(x)-f^\star} \le 2L^2(1+\sigma_1)^2$. The smoothness of $\hat{h}_{x,\xi}$ is $\frac{L\|\nabla f_\xi(x)\|^2}{\|\nabla f(x)\|^2} \le L(1+\sigma_1)^2$. The smoothness of $r_{x,\xi,\zeta}$ is $2L^2$. The smoothness of $h_{x,\xi,\zeta}$ is L.

For the Lipschitzness of $r_{x,\xi,\zeta}$ and $h_{x,\xi,\zeta}$,

$$\|\nabla h_{x,\xi,\zeta}(P)\|_F \le \tfrac{\|\nabla f_\zeta(x-P\nabla f_\xi(x)\|}{\|\nabla f_\xi(x)\|} \le \tfrac{\|\nabla f_\zeta(x-P\nabla f_\xi(x))-\nabla f_\zeta(x)\|}{\|\nabla f_\xi(x)\|} + \tfrac{\|\nabla f_\zeta(x)\|}{\|\nabla f_\xi(x)\|} \le LD + \tfrac{1+\sigma_1}{1-\sigma_1}$$

$$\|\nabla r_{x,\xi,\zeta}(P)\|_F \le \tfrac{\|\nabla f_\zeta(x-P\nabla f_\xi(x)\|}{f_\xi(x)-f^\star} \le \tfrac{2L\|\nabla f_\zeta(x-P\nabla f_\xi(x)\|}{\|\nabla f_\xi(x)\|} \le 2L\left(LD + \tfrac{1+\sigma_1}{1-\sigma_1}\right).$$

Therefore $r_{x,\xi,\zeta}$ is $2L\left(LD + \frac{1+\sigma_1}{1-\sigma_1}\right)$-Lipschitz and $h_{x,\xi,\zeta}$ is $\left(LD + \frac{1+\sigma_1}{1-\sigma_1}\right)$-Lipschitz. □

B.4.2 Proof of Lemma B.2

Before proving Lemma B.2, we introduce the Azuma's concentration inequality for martingale difference sequence (MDS).

Lemma B.7. Let $Z_1, \dots, Z_K$ be an MDS, if $|Z_k| \le B$ for all k almost surely, then for every $\delta \in (0,1)$,

$$\mathbb{P}\left(\textstyle\sum_{k=1}^K Z_k \ge B\sqrt{2K\log\frac{1}{\delta}}\right) \le \delta.$$

Proof of Lemma B.2. **Part (i) Ratio feedback, strongly convex.** Suppose there exists stepsize P and condition number $\kappa_\star \le \kappa$ such that,

$$\tfrac{1}{\kappa_\star}P^{-1} \preceq \nabla^2 f(x) \preceq P^{-1}, \qquad \forall x.$$

Let $P_\star^r = cP$, where $c \le 1$ is a constant to be chosen. Then the following properties hold,

$$\tfrac{c}{\kappa_\star}I \preceq P_\star^{1/2}\nabla^2 f(x) P_\star^{1/2} \preceq cI \quad \text{and} \quad \tfrac{c}{\kappa_\star\mu}I \preceq P_\star \preceq \tfrac{c}{L}I. \tag{17}$$

Consider the descent property,

$$\begin{aligned}
&f(x^k - P_\star^r\nabla f_{\xi^k}(x^k)) - f(x^k)\\
=&-\langle\nabla f(x^k), P_\star^r\nabla f_{\xi^k}(x^k)\rangle + \tfrac{1}{2}\langle P_\star^r\nabla f_{\xi^k}(x^k), \nabla^2 f(\psi^k)P_\star^r\nabla f_{\xi^k}(x^k)\rangle\\
\le&-\|\nabla f(x^k)\|^2_{P_\star^r} - \langle\nabla f_{\xi^k}(x^k) - \nabla f(x^k), P_\star^r\nabla f(x^k)\rangle + \tfrac{c}{2}\|\nabla f_{\xi^k}(x^k)\|^2_{P_\star^r}\\
=&-\left(1-\tfrac{c}{2}\right)\|\nabla f(x^k)\|^2_{P_\star^r} + \tfrac{c}{2}\|\nabla f_{\xi^k}(x^k) - \nabla f(x^k)\|^2_{P_\star^r} + (c-1)\langle\nabla f_{\xi^k}(x^k) - \nabla f(x^k), P_\star^r\nabla f(x^k)\rangle\\
\le&-\left(1-\tfrac{c}{2}\right)\|\nabla f(x^k)\|^2_{P_\star^r} + \tfrac{c^2\sigma_1^2}{2L}\|\nabla f_{\xi^k}(x^k) - \nabla f(x^k)\|_2^2 + (c-1)\langle\nabla f_{\xi^k}(x^k) - \nabla f(x^k), P_\star^r\nabla f(x^k)\rangle\\
\le&-\left(1-\tfrac{c}{2}-\tfrac{c\kappa_\star\sigma_1^2}{2\kappa}\right)\|\nabla f(x^k)\|^2_{P_\star^r} + \tfrac{(c-1)c}{L}\langle\nabla f_{\xi^k}(x^k) - \nabla f(x^k), \nabla f(x^k)\rangle
\end{aligned} \tag{18}$$

where the first equality is by Taylor expansion, the second inequality is by (17), the third equality expands $\|\nabla f_{\xi^k}(x^k)\|^2_{P_\star^r} = \|\nabla f(x^k) + (\nabla f_{\xi^k}(x^k) - \nabla f(x^k))\|^2_{P_\star^r}$, the fourth inequality uses $P_\star^r \preceq \frac{c}{L}I$ in (17), and the last inequality is by (17).

Since $\frac{c}{\kappa_\star} I \preceq {P_\star^r}^{1/2} \nabla^2 f(x) {P_\star^r}^{1/2}$,

$$f(x) - f^\star \leq \frac{\kappa_\star}{2c} \|\nabla f(x)\|^2_{P_\star}.$$

Then, substitute this inequality back into (18),

$$\begin{aligned} f(x^k - P_\star^r \nabla f_{\xi^k}(x^k)) - f(x^k) \leq & -\left(\frac{2c}{\kappa_\star} - \frac{c^2}{\kappa_\star} - \frac{c^2\sigma_1^2}{\kappa}\right)(f(x^k) - f^\star) \\ & + \frac{(c-1)c}{L}\langle \nabla f_{\xi^k}(x^k) - \nabla f(x^k), \nabla f(x^k)\rangle \end{aligned}$$

Choose $c = \frac{1}{1+(\kappa_\star/\kappa)\sigma_1^2}$, and rearrange,

$$\hat{r}_{x^k,\xi^k}(P_\star^r) \leq 1 - \frac{1}{\kappa_\star\left(1+\frac{\kappa_\star}{\kappa}\sigma_1^2\right)} - \frac{\frac{\kappa_\star}{\kappa}\sigma_1^2}{\left[1+\frac{\kappa_\star}{\kappa}\sigma_1^2\right]^2 L} \frac{\langle \nabla f_{\xi^k}(x^k) - \nabla f(x^k), \nabla f(x^k)\rangle}{f(x^k) - f(x^\star)}. \tag{19}$$

Define

$$Z_k = \frac{\langle \nabla f_{\xi^k}(x^k) - \nabla f(x^k), \nabla f(x^k)\rangle}{f(x^k) - f(x^\star)},$$

This is an MDS, and $|Z_k| \leq \frac{\sigma_1 \|\nabla f(x^k)\|^2}{f(x^k) - f(x^\star)} \leq 2L\sigma_1$ for all k. By Lemma B.7, w.p. at least $1 - \frac{\delta}{2}$,

$$\textstyle\sum_{k=1}^K Z_k \leq 2L\sigma_1\sqrt{2K\log\frac{2}{\delta}}. \tag{20}$$

Telescope (19) and use concentration inequality (20) and gets (13).

Part (ii) Hypergradient feedback, convex. Consider the descent property with scalar stepsize $P_\star^h = \alpha I$ where α is to be chosen,

$$\begin{aligned} & f_{\zeta^k}(x^k - \alpha \nabla f_{\xi^k}(x^k)) - f_{\zeta^k}(x^k) \\ \leq & -\alpha\langle \nabla f_{\xi^k}(x^k), \nabla f_{\zeta^k}(x^k)\rangle + \tfrac{\alpha^2 L}{2}\|\nabla f_{\xi^k}(x^k)\|^2 \\ = & -\alpha\langle \nabla f_{\xi^k}(x^k), \nabla f(x^k)\rangle - \alpha\langle \nabla f_{\xi^k}(x^k), \nabla f_{\zeta^k}(x^k) - \nabla f(x^k)\rangle + \tfrac{\alpha^2 L}{2}\|\nabla f_{\xi^k}(x^k)\|^2 \\ = & (-\alpha + \alpha^2 L)\langle \nabla f_{\xi^k}(x^k), \nabla f(x^k)\rangle + \tfrac{\alpha^2 L}{2}\|\nabla f_{\xi^k}(x^k) - \nabla f(x^k)\|^2 - \tfrac{\alpha^2 L}{2}\|\nabla f(x^k)\|^2 \\ & - \alpha\langle \nabla f_{\xi^k}(x^k), \nabla f_{\zeta^k}(x^k) - \nabla f(x^k)\rangle \\ = & \left(\tfrac{\alpha^2 L\sigma_1^2}{2} + \tfrac{\alpha^2 L}{2} - \alpha\right)\|\nabla f(x^k)\|^2 + (-\alpha + \alpha^2 L)\langle \nabla f_{\xi^k}(x^k) - \nabla f(x^k), \nabla f(x^k)\rangle \\ & - \alpha\langle \nabla f_{\xi^k}(x^k), \nabla f_{\zeta^k}(x^k) - \nabla f(x^k)\rangle \end{aligned}$$

where the first equality is due to,

$$\|\nabla f_{\xi^k}(x^k)\|^2 = \|\nabla f_{\xi^k}(x^k) - \nabla f(x^k)\|^2 - \|\nabla f(x^k)\|^2 + 2\langle \nabla f_{\xi^k}(x^k), \nabla f(x^k)\rangle.$$

Let $\alpha = \frac{1}{L(1+\sigma_1^2)}$, and rearrange,

$$\hat{h}_{x^k,\xi^k}(P_\star^h) \leq -\frac{1}{2L(1+\sigma_1^2)} - \frac{\sigma_1^2}{L(1+\sigma_1^2)^2}\frac{\langle \nabla f_{\xi^k}(x^k) - \nabla f(x^k), \nabla f(x^k)\rangle}{\|\nabla f(x^k)\|^2} - \frac{\langle \nabla f_{\xi^k}(x^k), \nabla f_{\zeta^k}(x^k) - \nabla f(x^k)\rangle}{L(1+\sigma_1^2)\|\nabla f(x^k)\|^2} \tag{21}$$

Let $Z_k^{(1)} = \frac{\langle \nabla f(x^k,\xi^k) - \nabla f(x^k), \nabla f(x^k)\rangle}{\|\nabla f(x^k)\|^2}$. This is an MDS, and $|Z_k^{(1)}| \leq \sigma_1$ for all k. By Lemma B.7, w.p. at least $1 - \frac{\delta}{2}$,

$$\textstyle\sum_{k=1}^K Z_k^{(1)} \leq \sigma_1\sqrt{2K\log\frac{2}{\delta}}. \tag{22}$$

Let $Z_k^{(2)} = \frac{\langle \nabla f_{\xi^k}(x^k), \nabla f_{\zeta^k}(x^k) - \nabla f(x^k)\rangle}{\|\nabla f(x^k)\|^2}$. This is an MDS, and $|Z_k^{(2)}| \leq \sigma_1(1+\sigma_1)$ for all k. By Lemma B.7, w.p. at least $1 - \frac{\delta}{2}$,

$$\textstyle\sum_{k=1}^K Z_k^{(2)} \leq \sigma_1(1+\sigma_1)\sqrt{2K\log\frac{2}{\delta}}. \tag{23}$$

Telescope (21) and use concentration inequality (22), (23) gets (14). □

B.4.3 Proof of Lemma B.4

Proof. **Part (i) Regret of ratio feedback.** By the convexity of $\hat{r}_{x^k,\xi^k}$, we have

$$\begin{aligned}\hat{r}_{x^k,\xi^k}(P_k)-\hat{r}_{x^k,\xi^k}(P_\star) &\le \langle \nabla \hat{r}_{x^k,\xi^k}(P_k), P_k-P_\star\rangle \\ &= \langle \nabla r_{x^k,\xi^k,\zeta^k}(P_k), P_k-P_\star\rangle + \langle \nabla \hat{r}_{x^k,\xi^k}(P_k)-\nabla r_{x^k,\xi^k,\zeta^k}(P_k), P_k-P_\star\rangle \\ &= \tfrac{1}{2\eta}\|P_k-P_\star\|_F^2 - \tfrac{1}{2\eta}\|P_{k+1}-P_\star\|_F^2 + \tfrac{\eta}{2}\|\nabla r_{x^k,\xi^k,\zeta^k}(P_k)\|_F^2 \\ &\quad + \langle \nabla \hat{r}_{x^k,\xi^k}(P_k)-\nabla r_{x^k,\xi^k,\zeta^k}(P_k), P_k-P_\star\rangle \end{aligned} \tag{24}$$

Consider the last term,

$$\begin{aligned}\nabla \hat{r}_{x^k,\xi^k}(P_k)-\nabla r_{x^k,\xi^k,\zeta^k}(P_k) &= \tfrac{[\nabla f(x^k-P\nabla f_{\xi^k}(x^k))-\nabla f_{\zeta^k}(x^k-P\nabla f_{\xi^k}(x^k))]\nabla f_{\xi^k}(x^k)^\top}{f_{\xi^k}(x^k)-f^\star} \\ &\quad + \tfrac{\nabla f(x^k-P\nabla f_{\xi^k}(x^k))\nabla f_{\xi^k}(x^k)^\top}{f_{\xi^k}(x^k)-f^\star}\left(\tfrac{f_{\xi^k}(x^k)-f^\star}{f(x^k)-f^\star}-1\right)\end{aligned}$$

Let $Z_k^{(1)} = \left\langle \tfrac{[\nabla f(x^k-P\nabla f_{\xi^k}(x^k))-\nabla f_{\zeta^k}(x^k-P\nabla f_{\xi^k}(x^k))]\nabla f_{\xi^k}(x^k)^\top}{f_{\xi^k}(x^k)-f^\star}, P_k-P_\star\right\rangle$, $\{Z_k^{(1)}\}$ is an MDS, and has a uniform bound,

$$|Z_k^{(1)}| \le \tfrac{\sigma_1\|\nabla f(x^k-P\nabla f_{\xi^k}(x^k))\|\|\nabla f_{\xi^k}(x^k)\|}{f_{\xi^k}(x^k)-f^\star}\|P_k-P_\star\|_F \le 2\sigma_1 LD\tfrac{\|\nabla f(x^k-P\nabla f_{\xi^k}(x^k))\|}{\|\nabla f_{\xi^k}(x^k)\|} \le 2\sigma_1 LD\left(LD+\tfrac{1}{1-\sigma_1}\right).$$

By Azuma's inequality (Lemma B.7), w.p. $1-\frac{\delta}{2}$,

$$\textstyle\sum_{k=1}^K Z_k^{(1)} \le 2\sigma_1 LD\left(LD+\frac{1}{1-\sigma_1}\right)\sqrt{2K\log\frac{2}{\delta}}.$$

Let $Z_k^{(2)} = \left\langle \tfrac{\nabla f(x^k-P\nabla f_{\xi^k}(x^k))\nabla f_{\xi^k}(x^k)^\top}{f_{\xi^k}(x^k)-f^\star}\left(\tfrac{f_{\xi^k}(x^k)-f^\star}{f(x^k)-f^\star}-1\right), P_k-P_\star\right\rangle$,

$$\begin{aligned}|Z_k^{(2)}| &\le \tfrac{\|\nabla f(x^k-P\nabla f_{\xi^k}(x^k))\|\|\nabla f_{\xi^k}(x^k)\|}{f_{\xi^k}(x^k)-f^\star}\left|\tfrac{f_{\xi^k}(x^k)-f^\star}{f(x^k)-f^\star}-1\right|\|P_k-P_\star\|_F \\ &\le 2\sigma_0 LD\tfrac{\|\nabla f(x^k-P\nabla f_{\xi^k}(x^k))\|}{\|\nabla f_{\xi^k}(x^k)\|} \le 2\sigma_0 LD\left(LD+\tfrac{1}{1-\sigma_1}\right).\end{aligned}$$

Therefore, $\sum_{k=1}^K Z_k^{(2)} \le 2\sigma_0 LD\left(LD+\frac{1}{1-\sigma_1}\right)K$. Telescope (24), w.p. $1-\frac{\delta}{2}$,

$$\begin{aligned}\textstyle\sum_{k=1}^K(\hat{r}_{x^k,\xi^k}(P_k)-\hat{r}_{x^k,\xi^k}(P_\star)) &\le \frac{1}{2\eta}\|P_1-P_\star\|_F^2 + 2\eta L^2\left(LD+\tfrac{1+\sigma_1}{1-\sigma_1}\right)^2 K + 2\sigma_0 LD\left(LD+\tfrac{1}{1-\sigma_1}\right)K \\ &\quad + 2\sigma_1 LD\left(LD+\tfrac{1}{1-\sigma_1}\right)\sqrt{2K\log\tfrac{2}{\delta}}.\end{aligned}$$

Let $\eta = \frac{\|P_1-P_\star\|_F}{2L\left(\frac{1+\sigma_1}{1-\sigma_1}+LD\right)\sqrt{K}}$, we derived the final regret bound (15).

Part (ii) Regret of hypergradient feedback. By the convexity of $\hat{h}_{x^k,\xi^k}$, we have

$$\begin{aligned}\hat{h}_{x^k,\xi^k,\zeta^k}(P_k)-\hat{h}_{x^k,\xi^k,\zeta^k}(P_\star) &\le \langle \nabla \hat{h}_{x^k,\xi^k}(P_k), P_k-P_\star\rangle \\ &= \langle \nabla h_{x^k,\xi^k,\zeta^k}(P_k), P_k-P_\star\rangle + \langle \nabla \hat{h}_{x^k,\xi^k,\zeta^k}(P_k)-\nabla \hat{h}_{x^k,\xi^k,\zeta^k}(P_k), P_k-P_\star\rangle \\ &\le \tfrac{1}{2\eta}\|P_k-P_\star\|_F^2 - \tfrac{1}{2\eta}\|P_{k+1}-P_\star\|_F^2 + \tfrac{\eta}{2}\|\nabla h_{x^k,\xi^k,\zeta^k}(P_k)\|_F^2 \\ &\quad + \langle \hat{h}_{x^k,\xi^k,\zeta^k}(P_k)-\nabla h_{x^k,\xi^k,\zeta^k}(P_k), P_k-P_\star\rangle \end{aligned} \tag{25}$$

For the last term,

$$\nabla \hat{h}_{x^k,\xi^k,\zeta^k}(P_k) - \nabla h_{x^k,\xi^k,\zeta^k}(P_k) = \frac{\nabla f_{\zeta^k}(x^k - P\nabla f_{\xi^k}(x^k))\nabla f_{\xi^k}(x^k)^\top}{\|\nabla f_{\xi^k}(x^k)\|^2}\left(1 - \frac{\|\nabla f_{\xi^k}(x^k)\|^2}{\|\nabla f(x^k)\|^2}\right)$$

Let $Z_k^{(2)} = \left(1 - \frac{\|\nabla f_{\xi^k}(x^k)\|^2}{\|\nabla f(x^k)\|^2}\right)\left\langle \frac{\nabla f_{\zeta^k}(x^k - P\nabla f_{\xi^k}(x^k))\nabla f_{\xi^k}(x^k)^\top}{\|\nabla f_{\xi^k}(x^k)\|^2}, P_k - P_\star \right\rangle$, and

$$|Z_k^{(2)}| \le (\sigma_1^2 + 2\sigma_1)D\left(LD + \tfrac{1}{1-\sigma_1}\right) \le 3\sigma_1 D\left(LD + \tfrac{1+\sigma_1}{1-\sigma_1}\right).$$

Therefore, $\sum_{k=1}^K Z_k^{(2)} \le 3\sigma_1 D\left(LD + \frac{1+\sigma_1}{1-\sigma_1}\right)K$. Telescope (25),

$$\sum_{k=1}^K (\hat{h}_{x^k,\xi^k}(P_k) - \hat{h}_{x^k,\xi^k}(P_\star)) \le \frac{1}{2\eta}\|P_1 - P_\star\|_F^2 + \frac{\eta}{2}\left(LD + \tfrac{1+\sigma_1}{1-\sigma_1}\right)^2 K + 3\sigma_1 D\left(LD + \tfrac{1+\sigma_1}{1-\sigma_1}\right)K.$$

Let $\eta = \frac{\|P_1 - P_\star\|_F}{\left(LD + \frac{1+\sigma_1}{1-\sigma_1}\right)\sqrt{K}}$, we derived the final regret bound (16). □

B.4.4 Proof of Proposition B.5

Proof of Proposition B.5. **Part (i) Ratio feedback, strongly convex.** By the definiton of ratio feedback,

$$\frac{f(x^K) - f(x^\star)}{f(x^0) - f(x^\star)} = \prod_{k=0}^{K-1} \hat{r}_{x^k,\xi^k}(P_k) \le \left(\frac{1}{K}\sum_{k=0}^{K-1} \hat{r}_{x^k,\xi^k}(P_k)\right)^K$$

Apply Lemma B.2 (i), Lemma B.4 (i) and by union bound, w.p. $1-\delta$,

$$\frac{f(x^K) - f(x^\star)}{f(x^0) - f(x^\star)} \le \left(1 - \frac{1}{\kappa_\star\left(1 + \frac{\kappa_\star}{\kappa}\sigma_1^2\right)} + 2\sigma_0 LD\left(LD + \tfrac{1}{1-\sigma_1}\right) + \frac{C}{\sqrt{K}}\right)^K$$

where $C = 2LD\left(LD + \frac{1+\sigma_1}{1-\sigma_1}\right) + 2\sigma_1 LD\left(LD + \frac{1}{1-\sigma_1}\right)\sqrt{2\log\frac{2}{\delta}}$. To ensure linear convergence, it suffices to impose $2\sigma_0 LD\left(LD + \frac{1}{1-\sigma_1}\right) \le \frac{1}{2(1+\sigma_1^2)\kappa_\star}$. Let $\sigma_1 \le \frac{1}{2}$, the condition reduced to $\sigma_0 \le \frac{1}{5LD(LD+2)\kappa_\star} = \mathcal{O}\left(\frac{1}{L^2D^2\kappa_\star}\right)$. Thus completes the proof of (i).

Part (ii) Hypergradient feedback, strongly convex. By null step and the definition of $\hat{h}_{x^k,\xi^k,\zeta^k}(P_k)$,

$$\begin{aligned} f_{\zeta^k}(x^{k+1}) - f_{\zeta^k}(x^k) &= \min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P_k), 0\}\|\nabla f(x^k)\|^2 \\ &\le 2\mu\min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P), 0\}(f(x^k) - f^\star) \end{aligned}$$

Thus we have,

$$\begin{aligned} f(x^{k+1}) - f(x^k) &\le 2\mu\min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P_k), 0\}(f(x^k) - f^\star) + f(x^{k+1}) - f_{\zeta^k}(x^{k+1}) + f_{\zeta^k}(x^k) - f(x^k) \\ &\le 2\mu\min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P_k), 0\}(f(x^k) - f^\star) + \sigma_0[(f(x^{k+1}) - f^\star) + (f(x^k) - f^\star)] \\ &\le \left(2\mu\min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P_k), 0\} + \tfrac{2\sigma_0}{1-\sigma_0}\right)(f(x^k) - f^\star) \end{aligned} \tag{26}$$

The last inequality is because,

$$\begin{aligned} f(x^{k+1}) - f(x^\star) &\le [1 + 2\mu\min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P_k), 0\}](f(x^k) - f^\star) + \sigma_0[f(x^{k+1}) - f^\star + f(x^k) - f^\star] \\ &\le (f(x^k) - f^\star) + \sigma_0[f(x^{k+1}) - f^\star + f(x^k) - f^\star], \end{aligned}$$

and when $\sigma_0 < 1$, $\frac{f(x^{k+1}) - f(x^\star)}{f(x^k) - f(x^\star)} \le \frac{1+\sigma_0}{1-\sigma_0}$. Combine with (26) and rearrange,

$$\frac{f(x^{k+1}) - f^\star}{f(x^k) - f^\star} \le 1 + 2\mu\min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P_k), 0\} + \tfrac{2\sigma_0}{1-\sigma_0}.$$

Telescope and by arithemetic mean inequality,

$$\begin{aligned}\frac{f(x^K)-f^\star}{f(x^0)-f^\star} &\le \prod_{k=0}^{K-1}\left(1+2\mu\min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P_k),0\}+\frac{2\sigma_0}{1-\sigma_0}\right)\\ &\le \left(\frac{1}{K}\sum_{k=0}^{K-1}\left(1+2\mu\min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P_k),0\}+\frac{2\sigma_0}{1-\sigma_0}\right)\right)^K\end{aligned}$$

By Lemma B.2 (ii) and Lemma B.4 (ii), w.p. $1-\delta$,

$$\frac{f(x^K)-f^\star}{f(x^0)-f^\star} \le \left(1+2\mu\min\left\{-\frac{1}{2L(1+\sigma_1^2)}+3\sigma_1 D\left(LD+\frac{1}{1-\sigma_1}\right),0\right\}+\frac{2\sigma_0}{1-\sigma_0}+\frac{C}{\sqrt{K}}\right)^K$$

where $C=\left[\frac{\sigma_1^3}{L(1+\sigma_1^2)^2}+\frac{\sigma_1(1+\sigma_1)}{L(1+\sigma_1^2)}\right]\sqrt{2\log\frac{1}{\delta}}+D\left(LD+\frac{1+\sigma_1}{1-\sigma_1}\right)$. To ensure linear convergence, it suffices to impose that,

$$3\sigma_1 D\left(LD+\frac{1}{1-\sigma_1}\right)\le\frac{1}{8L(1+\sigma_1^2)},\quad \frac{2\sigma_0}{1-\sigma_0}\le\frac{1}{8\kappa(1+\sigma_1^2)}.$$

With $\sigma_1\le\frac{1}{2}$, it suffices to let $\sigma_0\le\frac{1}{20\kappa+1},\sigma_1\le\frac{1}{30LD(LD+2)}$. Thus completes the proof of part (ii).

Part (iii) Hypergradient feedback, convex. Since $\sigma_0=0$, by the definition of $\hat{h}_{x^k,\xi^k,\zeta^k}(P_k)$,

$$\begin{aligned}f(x^{k+1})-f(x^k) &= \min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P_k),0\}\|\nabla f(x^k)\|^2\\ &\le \min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P_k),0\}\frac{[f(x^k)-f^\star]^2}{\|x^k-x^\star\|^2}\\ &\le \min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P_k),0\}\frac{[f(x^k)-f^\star]^2}{\Delta^2}\end{aligned}$$

where $\Delta:=\max_{x\in\{x:f(x)\le f(x^0)\}}\min_{x^\star\in\mathcal{X}^\star}\|x-x^\star\|$. Divide by $[f(x^{k+1})-f^\star][f(x^k)-f^\star]$ on both sides and by null step that $f(x^{k+1})\le f(x^k)$,

$$\frac{1}{f(x^k)-f^\star}-\frac{1}{f(x^{k+1})-f^\star}\le\frac{\min\{\hat{h}_{x^k,\xi^k,\zeta^k}(P_k),0\}}{\Delta^2}.$$

Telescope and rearrange,

$$f(x^K)-f^\star\le\min\left\{\frac{\Delta^2}{\max\left\{-\sum_{k=0}^{K-1}\hat{h}_{x^k,\xi^k,\zeta^k}(P_k),0\right\}},f(x^0)-f^\star\right\}$$

By Lemma B.2 (ii) and Lemma B.4 (ii), w.p. $1-\delta$,

$$f(x^K)-f^\star\le\min\left\{\frac{\Delta^2}{\max\left\{K\left(\frac{1}{2L(1+\sigma_1^2)}-3\sigma_1 D\left(LD+\frac{1+\sigma_1}{1-\sigma_1}\right)-\frac{C}{\sqrt{K}}\right),0\right\}},f(x^0)-f^\star\right\}$$

where $C=\left[\frac{\sigma_1^3}{L(1+\sigma_1^2)^2}+\frac{\sigma_1(1+\sigma_1)}{L(1+\sigma_1^2)}\right]\sqrt{2\log\frac{1}{\delta}}+D\left(LD+\frac{1+\sigma_1}{1-\sigma_1}\right)$. To get convergence, it suffices to impose $3\sigma_1 D\left(LD+\frac{1+\sigma_1}{1-\sigma_1}\right)\le\frac{1}{4L(1+\sigma_1^2)}$, $\sigma_1\le\frac{1}{15LD(LD+3)}=\mathcal{O}\left(\frac{1}{L^2D^2}\right)$. thus proved part (iii). □

C Proofs of results in Section 4

C.1 Proof of Lemma 4.2

Proof. By the definition of SVRG gradient estimator g_t, and L-smoothness,

$$\begin{aligned}\mathbb{E}_t[\|g_t\|^2] &= \mathbb{E}_t[\|\nabla f_{\xi^t}(x^t)-\nabla f_{\xi^t}(\tilde{x}^k)+\nabla f(\tilde{x}^k)\|^2]\\ &\le 2\mathbb{E}_t[\|\nabla f_{\xi^t}(x^t)-\nabla f_{\xi^t}(x^\star)\|^2]+2\mathbb{E}_t[\|\nabla f_{\xi^t}(\tilde{x}^k)-\nabla f_{\xi^t}(x^\star)-\nabla f(\tilde{x}^k)\|^2]\\ &\le 2\mathbb{E}_t[\|\nabla f_{\xi^t}(x^t)-\nabla f_{\xi^t}(x^\star)\|^2]+2\mathbb{E}_t[\|\nabla f_{\xi^t}(\tilde{x}^k)-\nabla f_{\xi^t}(x^\star)\|^2]\\ &\le 4L(f(x^t)-f^\star+f(\tilde{x}^k)-f^\star)\end{aligned}\tag{27}$$

Part (i) No momentum. The update rule is $x^{t+1} = x^t - cP_k g_t$. By $\underline{\alpha} I \preceq P_k \preceq \bar{\alpha} I$ and (27),

$$\begin{aligned}\mathbb{E}_t[f(x^{t+1})] &\le f(x^t) - c\nabla f(x^t)^\top P_k \nabla f(x^t) + \tfrac{c^2 L}{2}\mathbb{E}[\|P_k g_t\|^2] \\ &\le f(x^t) - c\,\underline{\alpha}\|\nabla f(x^t)\|^2 + \tfrac{c^2\bar{\alpha}^2 L}{2}\mathbb{E}[\|g_t\|^2] \\ &= f(x^t) - \left(2c\,\underline{\alpha}\,\mu - 2c^2\bar{\alpha}^2 L^2\right)(f(x^t) - f^\star) + 2c^2\bar{\alpha}^2 L^2 (f(\tilde{x}^k) - f^\star).\end{aligned}$$

Let $V^t := f(x^t) - f^\star$ proves (9).

Part (ii) Bounded momentum. Define $d^t := x^t - x^{t-1}$. Since f is L-smooth,

$$\begin{aligned}\mathbb{E}_t[f(x^{t+1})] &\le f(x^t) + \langle \nabla f(x^t), x^{t+1} - x^t\rangle + \tfrac{L}{2}\mathbb{E}_t[\|x^{t+1} - x^t\|^2] \\ &= f(x^t) + \langle \nabla f(x^t), -c\,\alpha\nabla f(x^t) + \beta_k d^t\rangle + \tfrac{L}{2}\mathbb{E}_t[\|d^{t+1}\|^2]. \end{aligned} \tag{28}$$

By the update rule, $d^{t+1} = -c\,\alpha g_t + \beta_k d^t$. Take conditional expectation and by (27),

$$\begin{aligned}\mathbb{E}_t[\|d^{t+1}\|^2] &= c^2\alpha^2\mathbb{E}_t[\|g_t\|^2] + \beta_k^2\|d^t\|^2 - 2c\,\alpha\beta_k\langle\nabla f(x^t), d^t\rangle \\ &\le 4c^2\alpha^2 L(f(x^t) - f^\star) + \beta_k^2\|d^t\|^2 - 2c\,\alpha\beta_k\langle\nabla f(x^t), d^t\rangle + 4c^2\alpha^2 L(f(\tilde{x}^k) - f^\star). \end{aligned} \tag{29}$$

Let $V^t = f(x^t) - f^\star + \frac{L}{2}\|d^t\|^2$, by (28) and (29),

$$\begin{aligned}\mathbb{E}_t[V^{t+1}] &\le f(x^t) - f^\star + \langle\nabla f(x^t), -c\,\alpha\nabla f(x^t) + \beta_k d^t\rangle + L\mathbb{E}_t[\|d^{t+1}\|^2] \\ &\le f(x^t) - f^\star - c\,\alpha\|\nabla f(x^t)\|^2 + (1 - 2c\,\alpha L)\beta_k\langle\nabla f(x^t), d^t\rangle + L\beta_k^2\|d^t\|^2 \\ &\quad + 4c^2\alpha^2 L^2(f(x^t) - f^\star) + 4c^2\alpha^2 L^2(f(\tilde{x}^k) - f^\star) \\ &= V^t - \left(\tfrac{1}{2} - \beta_k^2\right)L\|d^t\|^2 - c\,\alpha\|\nabla f(x^t)\|^2 + 4c^2\alpha^2 L^2(f(\tilde{x}^k) - f^\star) \\ &\quad + (1 - 2c\,\alpha L)\beta_k\langle\nabla f(x^t), d^t\rangle + 4c^2\alpha^2 L^2(f(x^t) - f^\star) \\ &\le V^t - \left(\tfrac{1}{2} - \beta_k^2\right)L\|d^t\|^2 - (c\,\alpha - 2c^2\alpha^2 L\kappa)\|\nabla f(x^t)\|^2 \\ &\quad + (1 - 2c\,\alpha L)\beta_k\langle\nabla f(x^t), d^t\rangle + 4c^2\alpha^2 L^2(f(\tilde{x}^k) - f^\star).\end{aligned}$$

Thus completes the proof of (10). □

C.2 Proof of Proposition 4.4

Proof. **Part (i)** `SVRG`. Let $c = 1$ and $\alpha = \bar{\alpha} = \underline{\alpha}$ in (9) and telescope,

$$\begin{aligned}\mathbb{E}[f(x^m)] &\le f(x^0) + 2m\alpha^2 L^2(f(\tilde{x}^k) - f^\star) - 2(\alpha\mu - \alpha^2 L^2)\left[\sum_{t=0}^{m-1}(\mathbb{E}[f(x^t)] - f(x^\star))\right] \\ &= f(\tilde{x}^k) + 2m\alpha^2 L^2(f(\tilde{x}^k) - f^\star) - 2m(\alpha\mu - \alpha^2 L^2)\mathbb{E}[f(\tilde{x}^{k+1}) - f^\star].\end{aligned}$$

Since $\alpha < \frac{1}{\kappa L}$, then $\alpha\mu - \alpha^2 L^2 > 0$. Rearrange and the contraction ratio is,

$$\tfrac{\mathbb{E}[f(\tilde{x}^{k+1}) - f^\star]}{f(\tilde{x}^k) - f^\star} \le \tfrac{1 + 2m\alpha^2 L^2}{2m(\alpha\mu - \alpha^2 L^2)}. \tag{30}$$

Telescope (30) proves (i). If $\alpha = \frac{1}{4\kappa L}$, the contraction ratio is $\frac{1}{3} + \frac{4\kappa^2}{3m}$.

Part (ii) `SVRG-HBM`. Let $c = 1$ and $\beta_k = \beta \le \bar{\beta}$ in (10). Suppose $\alpha < \frac{1}{2\kappa L}$ and $\bar{\beta} \le \sqrt{\alpha L - 2\alpha^2 L^2\kappa}$,

$$\begin{aligned}\mathbb{E}_t[V^{t+1}] &\le V^t - \left(\tfrac{1}{2} - \bar{\beta}^2\right)L\|d^t\|^2 - (\alpha - 2\alpha^2 L\kappa)\|\nabla f(x^t)\|^2 \\ &\quad + (1 - 2\alpha L)\bar{\beta}\langle\nabla f(x^t), d^t\rangle + 4\alpha^2 L^2(f(\tilde{x}^k) - f^\star) \\ &\le V^t - \tfrac{L}{4}\|d^t\|^2 - (\alpha - 2\alpha^2 L\kappa)\|\nabla f(x^t)\|^2 + \bar{\beta}|\langle\nabla f(x^t), d^t\rangle| + 4\alpha^2 L^2(f(\tilde{x}^k) - f^\star) \\ &\le V^t - \left(\alpha - 2\alpha^2 L\kappa - \tfrac{\bar{\beta}^2}{L}\right)\|\nabla f(x^t)\|^2 + 4\alpha^2 L^2(f(\tilde{x}^k) - f^\star) \\ &\le V^t - \left(2\alpha\mu - 4\alpha^2 L^2 - \tfrac{2\bar{\beta}^2}{\kappa}\right)(f(x^t) - f^\star) + 4\alpha^2 L^2(f(\tilde{x}^k) - f^\star)\end{aligned}$$

$$\frac{\mathbb{E}[f(\tilde{x}^{k+1})-f^\star]}{f(\tilde{x}^k)-f^\star} \leq \frac{1+4m\alpha^2L^2}{m\left(2\alpha\mu-4\alpha^2L^2-\frac{2\bar{\beta}^2}{\kappa}\right)} \tag{31}$$

Telescope (31) proves (ii). If $\alpha=\frac{1}{8\kappa L}$ and $\bar{\beta}=\frac{1}{\sqrt{32}\kappa}$, the contraction ratio is $\frac{8\kappa^2}{m}+\frac{1}{2}$. □

C.3 Proof of Proposition 4.6

Proof. **Part (i) Scalar Stepsize.** For hypergradient feedback, the update rule is,

$$\alpha_{k+1}=\alpha_k+\eta\frac{\nabla f(x^k-\alpha_k\nabla f(x^k))^\top\nabla f(x^k)}{\|\nabla f(x^k)\|^2},$$

where x could be arbitrary. Let $x^{k+1}=x^k-\alpha_k\nabla f(x^k)$, since ∇f is μ-strongly monotone and L-Lipschitz,

$$\mu\alpha_k^2\|\nabla f(x^k)\|^2\leq\langle x^k-x^{k+1},\nabla f(x^k)-\nabla f(x^{k+1})\rangle\leq L\alpha_k^2\|\nabla f(x^k)\|^2 \tag{32}$$

Since $\nabla f(x^{k+1})^\top\nabla f(x^k)=\|\nabla f(x^k)\|^2-\frac{1}{\alpha_k}\langle x^k-x^{k+1},\nabla f(x^k)-\nabla f(x^{k+1})\rangle$, we have

$$1-L\alpha_k\leq\frac{\nabla f(x^{k+1})^\top\nabla f(x^k)}{\|\nabla f(x^k)\|^2}\leq 1-\mu\alpha_k$$

Finally, by $\alpha_k\in\left[\frac{1}{L},\frac{1}{\mu}\right]$, we derived the bound on α_{k+1},

$$\frac{1}{L}\leq(1-\eta L)\alpha_k+\eta\leq\alpha_k+\eta\frac{\nabla f(x^{k+1})^\top\nabla f(x^k)}{\|\nabla f(x^k)\|^2}\leq(1-\eta\mu)\alpha_k+\eta\leq\frac{1}{\mu}.$$

For ratio feedback, the update rule for scalar stepsize is,

$$\begin{aligned}\alpha_{k+1}&=\alpha_k+\eta\frac{\nabla f(x^k-\alpha_k\nabla f(x^k))^\top\nabla f(x^k)}{f(x^k)-f^\star}\\&=\alpha_k+\eta\frac{\|\nabla f(x^k)\|^2-\frac{1}{\alpha_k}\langle x^k-x^{k+1},\nabla f(x^k)-\nabla f(x^{k+1})\rangle}{f(x^k)-f^\star}\end{aligned}$$

By inequality (32),

$$\alpha_k+\eta\frac{(1-L\alpha_k)\|\nabla f(x^k)\|^2}{f(x^k)-f^\star}\leq\alpha_{k+1}\leq\alpha_k+\eta\frac{(1-\mu\alpha_k)\|\nabla f(x^k)\|^2}{f(x^k)-f^\star}$$

Since $\alpha_k\in\left[\frac{1}{L},\frac{1}{\mu}\right]$, $(1-L\alpha_k)\leq 0$ and $(1-\mu\alpha_k)\geq 0$. Also, by $\frac{\|\nabla f(x^k)\|^2}{f(x^k)-f^\star}\leq 2L$,

$$(1-2\eta L^2)\alpha_k+2\eta L\leq\alpha_{k+1}\leq(1-2\eta L\mu)\alpha_k+2\eta L.$$

When $\eta\leq\frac{1}{2L^2}$, we have $\alpha_{k+1}\in\left[\frac{1}{L},\frac{1}{\mu}\right]$,

$$\frac{1}{L}=(1-2\eta L^2)\frac{1}{L}+2\eta L\leq\alpha_{k+1}\leq(1-2\eta L\mu)\frac{1}{\mu}+2\eta L=\frac{1}{\mu}.$$

Part (ii) Matrix Stepsize. For hypergradient feedback, the update rule of matrix stepsize is,

$$\begin{aligned}\|P_{k+1}\|_F&=\left\|(1-\eta\rho)P_k+\eta\frac{\nabla f(x^k-P_k\nabla f(x^k))\nabla f(x^k)^\top}{\|\nabla f(x^k)\|^2}\right\|_F\\&\leq(1-\eta\rho)\|P_k\|_F+\eta\frac{\|\nabla f(x^k-P_k\nabla f(x^k))\|}{\|\nabla f(x^k)\|}\\&\leq(1-\eta\rho)\|P_k\|_F+\eta(1+L\|P_k\|_F)\\&=(1-(\rho-L)\eta)\|P_k\|_F+\eta\\&\leq(1-(\rho-L)\eta)^{k+1}\|P_0\|_F+\frac{1}{\rho-L}\end{aligned}$$

where the first inequality is by triangular inequality and the property of Fronenius norm, the second inequality is by the L-smoothness of f, and the last inequality is by geometric sum. Given the parameter choice $\rho = 2L$, $P_0 = 0$, and $\eta \le \frac{1}{L}$, we have part (ii) hold by $\|P_k\|_2 \le \|P_k\|_F \le \frac{1}{L}$.

For ratio feedback, the update rule of matrix stepsize is,

$$\begin{aligned}
\|P_{k+1}\|_F &= \left\|(1-\eta\rho)P_k + \eta\tfrac{\nabla f(x^k - P_k\nabla f(x^k))\nabla f(x^k)^\top}{f(x^k)-f^\star}\right\|_F \\
&\le (1-\eta\rho)\|P_k\|_F + \eta\tfrac{\|\nabla f(x^k - P_k\nabla f(x^k))\|\|\nabla f(x^k)\|}{f(x^k)-f^\star} \\
&\le (1-\eta\rho)\|P_k\|_F + 2\eta L\tfrac{\|\nabla f(x^k - P_k\nabla f(x^k))\|}{\|\nabla f(x^k)\|} \\
&\le (1-\eta\rho)\|P_k\|_F + 2\eta L(1 + L\|P_k\|_F) \\
&= (1-(\rho-2L^2)\eta)\|P_k\|_F + 2\eta L \\
&\le (1-(\rho-2L^2)\eta)^{k+1}\|P_0\|_F + \tfrac{2L}{\rho-2L^2}
\end{aligned}$$

Let $\rho = 4L^2$, $P_0 = 0$, and $\eta \le \frac{1}{2L^2}$, we have part (ii) hold by $\|P_k\|_2 \le \|P_k\|_F \le \frac{1}{L}$. □

C.4 Proof of Theorem 4.8

Proof. **Part (i) No momentum, scalar stepsize.** The update rule is $x^{t+1} = x^t - c\alpha_k g_t$. Let $\bar{\alpha} = \underline{\alpha} = \alpha_k$ in (9) from Lemma 4.2,

$$\mathbb{E}_t[V^{t+1}] \le V^t - (2c\alpha_k\mu - 2c^2\alpha_k^2L^2)(f(x^t) - f^\star) + 2c^2\alpha_k^2L^2(f(\tilde{x}^k) - f^\star).$$

Suppose $c \le \frac{1}{\alpha_k L\kappa}$, such that $2c\alpha_k\mu - 2c^2\alpha_k^2L^2 \ge 0$. Telescope and rearrange, then the contraction factor is,

$$\tfrac{\mathbb{E}[f(\tilde{x}^{k+1})-f^\star]}{f(\tilde{x}^k)-f^\star} \le \tfrac{1+2mc^2\alpha_k^2L^2}{2m(c\alpha_k\mu - c^2\alpha_k^2L^2)} = \tfrac{1}{2m(c\alpha_k\mu-c^2\alpha_k^2L^2)} + \tfrac{c^2\alpha_k^2L^2}{(c\alpha_k\mu-c^2\alpha_k^2L^2)}. \tag{33}$$

From Proposition 4.6 (i), the stepsize is bounded $\alpha_k \in \left[\frac{1}{L}, \frac{1}{\mu}\right]$, then $c \le \frac{1}{\alpha_k L\kappa}$ implies that $c \le \frac{1}{\kappa^2}$. Suppose $c \le \frac{1}{\kappa(\kappa+1)}$, we have the first term of (33) is maximized at $\alpha_k = \frac{1}{L}$, and the second term of (33) is maximized at $\alpha_k = \frac{1}{\mu}$. Substitute into (33) gets the upper bound of contraction factor,

$$\tfrac{\mathbb{E}[f(\tilde{x}^{k+1})-f^\star]}{f(\tilde{x}^k)-f^\star} \le \tfrac{\kappa}{2m(c-c^2\kappa)} + \tfrac{c\kappa^2}{1-c\kappa^2}.$$

If $c = \frac{1}{3\kappa^2}$, we can choose $m = 9\kappa^3$ to have contraction factor $\frac{3}{4}$.

Part (ii) No momentum, matrix stepsize. The update rule is $x^{t+1} = x^t - cP_k g_t$. From Proposition 4.6 (ii) and the definition of $\mathcal{P}$, we have $\underline{\alpha}I \preceq P_k \preceq \frac{1}{L}I$. Let $\underline{\alpha}, \bar{\alpha} = \frac{1}{L}$ in (9) from Lemma 4.2 (i),

$$\mathbb{E}_t[V^{t+1}] \le V^t - \left(2c\underline{\alpha}\mu - 2c^2\right)(f(x^t) - f^\star) + 2c^2(f(\tilde{x}^k) - f^\star).$$

Suppose $c \le \underline{\alpha}\mu$ such that $2c\underline{\alpha}\mu - 2c^2 \ge 0$, then the contraction factor is,

$$\tfrac{\mathbb{E}[f(\tilde{x}^{k+1})-f^\star]}{f(\tilde{x}^k)-f^\star} \le \tfrac{1+2mc^2}{2m(c\underline{\alpha}\mu-c^2)} = \tfrac{1}{2m(c\underline{\alpha}\mu-c^2)} + \tfrac{c}{(\underline{\alpha}\mu-c)}.$$

If $c = \frac{\underline{\alpha}\mu}{3}$, we can choose $m = \frac{9}{\underline{\alpha}^2\mu^2}$ to have contraction factor $\frac{3}{4}$.

Part (iii) Bounded momentum, scalar stepsize. The update rule is $x^{t+1} = x^t - c\alpha_k g_t + \beta_k(x^t - x^{t-1})$. From the proof of Lemma 4.2, suppose $c\alpha_k < \frac{1}{2\kappa L}$ and $0 \le \beta_k \le \bar{\beta}$, we have

$$\mathbb{E}_t[V^{t+1}] \le V^t - \left(2c\alpha_k\mu - 4c^2\alpha_k^2L^2 - \tfrac{2\bar{\beta}^2}{\kappa}\right)(f(x^t) - f^\star) + 4c^2\alpha_k^2L^2(f(\tilde{x}^k) - f^\star).$$

Suppose $2\bar{\beta}^2 \leq c\alpha_k\kappa\mu$ and $c\alpha_k\mu - 4c^2\alpha_k^2 L^2 \geq 0$, then the contraction factor is,

$$\frac{\mathbb{E}[f(\tilde{x}^{k+1})-f^\star]}{f(\tilde{x}^k)-f^\star} \leq \frac{1+4mc^2\alpha_k^2L^2}{m(c\alpha_k\mu-4c^2\alpha_k^2L^2)} = \frac{1}{m(c\alpha_k\mu-4c^2\alpha_k^2L^2)} + \frac{4c\alpha_k L^2}{\mu-4c\alpha_k L^2}. \tag{34}$$

From Proposition 4.6 (i), the stepsize is bounded $\alpha_k \in \left[\frac{1}{L}, \frac{1}{\mu}\right]$. Suppose $c \leq \frac{1}{4\kappa(\kappa+1)}$, we have the first term of (34) is maximized at $\alpha_k = \frac{1}{L}$, and the second term of (34) is maximized at $\alpha_k = \frac{1}{\mu}$. Substitute into (34) gets the upper bound of contraction factor,

$$\frac{\mathbb{E}[f(\tilde{x}^{k+1})-f^\star]}{f(\tilde{x}^k)-f^\star} \leq \frac{\kappa}{m(c-c^2\kappa)} + \frac{4c\kappa^2}{1-4c\kappa^2}.$$

If $c = \frac{1}{12\kappa^2}$, which implies that $\bar{\beta} \leq \frac{1}{2\sqrt{6}\kappa}$, we choose $m = 72\kappa^3$ to have contraction factor $\frac{3}{4}$. □

D Experimental details

D.1 Dataset details

Table 3 lists the details of 47 datasets used in Section 5, where n is the number of samples, and d is the number of features.

Dataset	n	d	Dataset	n	d
a1a	1,605	123	a2a	2,265	123
a3a	3,185	123	a4a	4,781	123
a5a	6,414	123	a6a	11,220	123
a7a	16,100	123	a8a	22,696	123
a9a	32,561	123	covtype	464,809	54
german.numer	800	24	gisette	6,000	5,000
ijcnn1	35,000	22	madelon	2,000	500
mushrooms	6,499	112	news20	15,996	1,355,191
phishing	8,844	68	rcv1	20,242	47,236
real-sim	57,847	20,958	splice	1,000	60
sonar	166	60	svmguide3	1,243	22
w1a	2,477	300	w2a	3,470	300
w3a	4,912	300	w4a	7,366	300
w5a	9,888	300	w6a	17,188	300
w7a	24,692	300	w8a	49,749	300
webspam	280,000	254	e2006	16,087	150,360
yearpredictionmsd	463,715	90	santander	160,000	200
miniboone	104,051	50	guillermo	16,000	4,296
creditcard	227,845	29	acsincome	1,331,600	11
medical	48,971	18	airlines	800,000	6
click-prediction	1,597,928	11	mtp	3,560	202
elevators	13,279	18	ailerons	11,000	40
superconduct	17,010	79	sarcos	39,146	21
jannis	46,064	54			

Table 3: Details of 47 datasets used in Section 5.

D.2 Practical variants

The difference between Practical `OSGM-SGD` (Algorithm 5) and `OSGM-SGD` (Algorithm 2) is in the definition of feedback functions. Practical `OSGM-SGD` uses in-sample feedback (3) while `OSGM-SGD` uses out-of-sample feedback (4). Also, practical `OSGM-SGD` drops the null step for efficiency.

Algorithm 5 Practical `OSGM-SGD`

Input: Initial iterate x^0, initial stepsize P_0, candidate stepsize set $\mathcal{P}$, learning rate η, feedback $\ell_{x,\xi,\zeta} \in \{r_{x,\xi,\zeta},\, h_{x,\xi,\zeta}\}$ defined by (3)
for $k = 0, 1, 2, \ldots$ **do**
  Sample ξ^k uniformly
  $x^{k+1} = x^k - P_k \nabla f_{\xi^k}(x^k)$
  $P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta \nabla \ell^k_{x^k,\xi^k,\zeta^k}(P_k)]$
end for

The implementation of practical variants `OSGM-SVRG` (Algorithm 6) and `OSGM-SketchySVRG` (Algorithm 4) are inspired by `OSGM-Best` introduced by [15], which tune the stepsize and momentum simultaneously,

$$x^+ = x - P\nabla f(x) + \beta(x - x^-).$$

Specifically, [15] introduces a joint hypergradient feedback,

$$h_{x,x^-}(P, \beta) = \frac{\phi_\omega(x^+(P,\beta), x) - \phi_\omega(x, x^-)}{\|\nabla f(x)\|^2 + \frac{\tau}{2}\|x - x^-\|^2},$$

where $\phi_\omega(x, x^-) := f(x) - f^\star + \frac{\omega}{2}\|x - x^-\|^2$ is the potential function for heavy-ball momentum. Then the stepsize and momentum are updated by online gradient descent. In `OSGM-SVRG` and `OSGM-SketchySVRG`, we substitute the deterministic gradients by their stochastic counterparts.

Algorithm 6 Practical `OSGM-SVRG`

Input: Initial $\tilde{x}^0$, $P_0 = 0$, $\beta_0 = 0$, epoch length m, `OSGM` learning rates η_P and η_β, candidate set of diagonal stepsize and momentum $\mathcal{P}$, $\mathcal{B}$
for $k = 0, 1, 2, \ldots$ **do**
  Compute snapshot gradient $\nabla f(\tilde{x}^k) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\tilde{x}^k)$
  Set $x^0 = \tilde{x}^k$
  for $t = 0, \ldots, m-1$ **do**
    Sample ξ^t uniformly
    Compute variance reduced gradient $g_t = \nabla f_{\xi^t}(x^t) - \nabla f_{\xi^t}(\tilde{x}^k) + \nabla f(\tilde{x}^k)$
    $x^{t+1} = x^t - P_t g_t + \beta_t(x^t - x^{t-1})$
    Define feedback $\ell_t(P, \beta) = \frac{f(x^t - P g_t + \beta(x^t - x^{t-1})) - f(x^t)}{\|g_t\|^2 + \|x^t - x^{t-1}\|^2}$
    Update stepsize: $P_{t+1} = \Pi_{\mathcal{P}}\big[P_t - \eta_P \nabla_P \ell_t(P_t, \beta_t)\big]$
    Update momentum: $\beta_{t+1} = \Pi_{\mathcal{B}}\big[\beta_t - \eta_\beta \nabla_\beta \ell_t(P_t, \beta_t)\big]$
  end for
  Choose $\tilde{x}^{k+1}$ uniformly from $\{x^0, \ldots, x^m\}$, set $P_0 = P_m$, $\beta_0 = \beta_m$
end for

D.3 Additional experiments

D.3.1 Evaluations of Algorithm 3

We evaluate the performance of `OSGM-SVRG` (Algorithm 3) on logistic regression problems. The batchsize is 256.

Benchmark algorithms. We benchmark the following variance reduction algorithms.

- `SVRG` [32] with updated frequency $m = \lceil n/256 \rceil$ and tuned stepsize.
- `SAGA` [18] with tuned stepsize.
- `L-Katyusha` [34]. `Loopless Katyusha` [1] with tuned stepsize.
- `OSGM-SVRG` with scalar stepsize, default `OSGM` learning rate $\eta = \frac{1}{L}$ and tuned momentum $\beta \in \{0.0, 0.9, 0.98\}$

Performance plots. Figure 5 shows the suboptimality plots on 8 medium-sized logistic regression problems. Notice that `OSGM-SVRG` with default learning rate outperforms baseline algorithms with tuned stepsize, especially in the later iterations. This experiment suggests the effectiveness of `OSGM-SVRG` as an outer-loop stepsize scheduler.

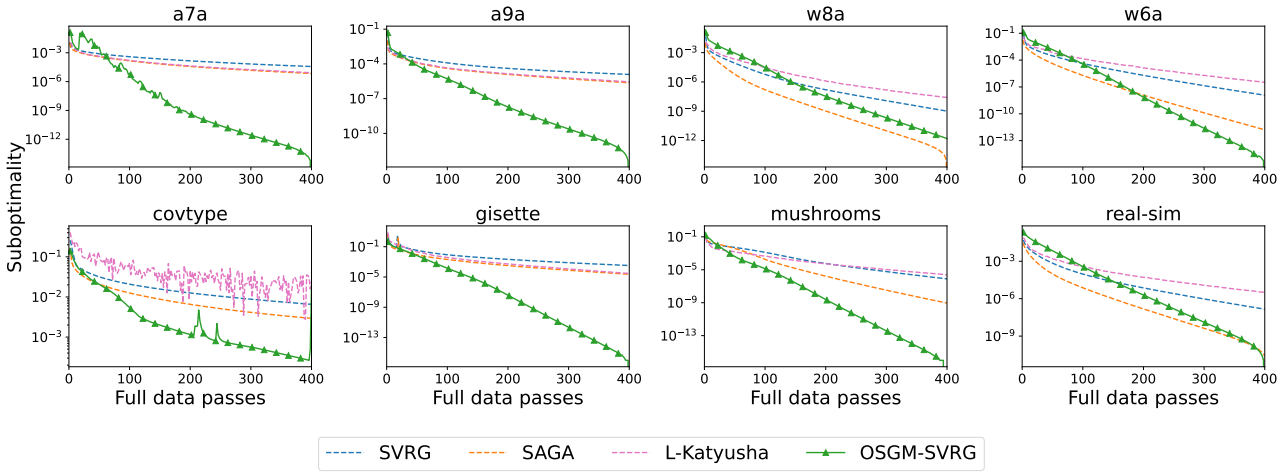

Figure 5: Performance of Algorithm 3 on logistic regression problems.

D.3.2 Evaluations of Algorithm 2

We evaluate the performance of Algorithm 2 with feedback (3) (labeled as Naive) and feedback (4) (labeled as Theory). The experiment setting is the same as Section 5. From the performance plot Figure 6, `OSGM-SGD` outperforms `SGD`, and `OSGM-SGD` outperforms `OSGM-SGD`. As suggested in Section 3, `OSGM-SGD` (3) may not converge in some cases.

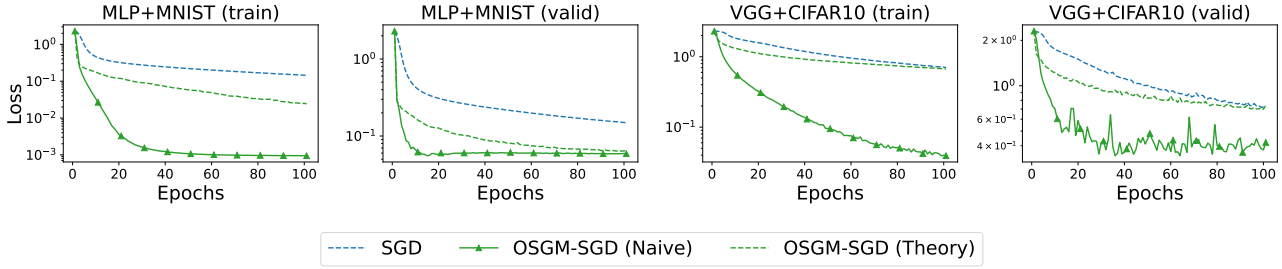

Figure 6: Performance of Algorithm 2 with feedback (3) and feedback (4).