
One Method to Rule Them All: Variance Reduction for Data, Parameters and Many New Methods

Filip Hanzely

KAUST

Thuwal, Saudi Arabia

filip.hanzely@kaust.edu.sa

Peter Richtárik*

KAUST

Thuwal, Saudi Arabia

peter.richtarik@kaust.edu.sa

Abstract

We propose a remarkably general variance-reduced method suitable for solving regularized empirical risk minimization problems with either a large number of training examples, or a large model dimension, or both. In special cases, our method reduces to several known and previously thought to be unrelated methods, such as SAGA [3], LSVRG [12, 15], JacSketch [8], SEGA [10] and ISEGA [21], and their arbitrary sampling and proximal generalizations. However, we also highlight a large number of new specific algorithms with interesting properties. We provide a single theorem establishing linear convergence of the method under smoothness and quasi strong convexity assumptions. With this theorem we recover best-known and sometimes improved rates for known methods arising in special cases. As a by-product, we provide the first unified method and theory for stochastic gradient and stochastic coordinate descent type methods.

1 Introduction

In this work we are studying stochastic algorithms for solving regularized empirical risk minimization problems, i.e., optimization problems of the form

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{j=1}^n f_j(x) + \psi(x), \quad (1)$$

We assume that the functions $f_j : \mathbb{R}^d \rightarrow \mathbb{R}$ are smooth and convex, and $\psi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ is a proper, closed and convex regularizer, admitting a cheap proximal operator. We write $f := \frac{1}{n} \sum_j f_j$.

Proximal gradient descent. A baseline method for solving problem (1) is (*proximal*) *gradient descent* (GD). This method performs a gradient step in f , followed by a proximal step² in ψ , i.e.,

$$x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha \nabla f(x^k)), \quad (2)$$

where $\alpha > 0$ is a stepsize. GD performs well when both n and d are not too large. However, in the big data (large n) and/or big parameter (large d) case, the formation of the gradient becomes overly expensive, rendering GD inefficient in both theory and practice. A typical remedy is to replace

*Also affiliated with the Moscow Institute of Physics and Technology, Dolgoprudny, Russia.

²The proximal operator is defined via $\text{prox}_{\alpha\psi}(x) := \text{argmin}_{u \in \mathbb{R}^d} \{\alpha\psi(u) + \frac{1}{2}\|u - x\|^2\}$.

the gradient by a cheap-to-compute random approximation. Typically, one replaces $\nabla f(x^k)$ with a random vector g^k whose mean is the gradient: $\mathbb{E}[g^k] = \nabla f(x^k)$, i.e., with a stochastic gradient. This results in the (*proximal*) *stochastic gradient descent* (SGD) method:

$$x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k). \quad (3)$$

Below we comment on the typical approaches to constructing g^k in the big n and big d regimes.

Proximal SGD. In the big n regime, the simplest choice is to set

$$g^k = \nabla f_j(x^k) \quad (4)$$

for an index $j \in [n] := \{1, 2, \dots, n\}$ chosen uniformly at random. By construction, it is n times cheaper to compute this estimator than the gradient, which is a key driving force behind the efficiency of this variant of SGD. However, there is an infinite array of other possibilities of constructing an unbiased estimator [24, 5]. Depending on how g^k is formed, (3) specializes to one of the many existing variants of proximal SGD, each with different convergence properties and proofs.

Proximal RCD. In the big d regime (this is interesting even if $n = 1$), the simplest choice is to set

$$g^k = d \langle \nabla f(x^k), e_i \rangle e_i, \quad (5)$$

where $\langle x, y \rangle = \sum_i x_i y_i$ is the standard Euclidean inner product, e_i is the i -th standard unit basis vector in \mathbb{R}^d , and i is chosen uniformly at random from $[d] := \{1, 2, \dots, d\}$. With this estimator, (3) specializes to (proximal) randomized coordinate decent (RCD). There are situations where it is d times cheaper to compute the partial derivative $\nabla_i f(x^k) := \langle \nabla f(x^k), e_i \rangle$ than the gradient, which is a key driving force behind the efficiency of RCD [26]. However, there is an infinite array of other possibilities for constructing an unbiased estimator of the gradient in a similar way [32, 33, 29].

Issues. For the sake of argument in the rest of this section, assume that f is a σ -strongly convex function, and let x^* be the (necessarily) unique solution of (1). It is well known that in this case, method (3) with estimator g^k defined as in (4) does *not* in general converge to x^* . Instead, SGD converges linearly to a neighborhood of x^* of size proportional to the stepsize α , noise $\nu^2 := \frac{1}{n} \sum_j \|\nabla f_j(x^*)\|^2$, and inversely proportional to σ [22, 23]. In the generic regime with $\nu^2 > 0$, the neighbourhood is nonzero, causing issues with convergence. This situation does not change even when tricks such as *mini-batching* or *importance sampling* (or a combination of both) are applied [25, 24, 5]. While these tricks affect both the (linear convergence) rate and the size of the neighbourhood, they are incapable³ of ensuring convergence to the solution. However, a remedy does exist: the situation with non-convergence can be resolved by using one of the many *variance-reduction* strategies for constructing g^k developed over the last several years [35, 3, 13, 20, 36]. Further, while it is well known that method (3) with estimator g^k defined as in (5) (i.e., randomized coordinate descent) converges to x^* for $\psi \equiv 0$ [26, 31, 32], it is also known that it does *not* generally converge to x^* unless the regularizer ψ is separable (e.g., $\psi(x) = \|x\|_1$ or $\psi(x) = c_1 \|x\|_1 + c_2 \|x\|_2^2$). In [10], an alternative estimator (known as SEGA) was constructed from the same (random) partial derivative information $\nabla f_i(x^k)$, one that does not suffer from this incompatibility with general regularizers ψ . This work resolved a long standing open problem in the theory of RCD methods.

2 Contributions

Having experienced a ‘‘Cambrian explosion’’ in the last 10 years, the world of efficient SGD methods is remarkably complex. There is a large and growing set of rules for constructing the gradient estimators g^k , with differing levels of sophistication and varying theoretical and practical properties.

³Unless, of course, in the special case when one uses the full batch approximation $g^k = \nabla f(x^k)$.

It includes the classical estimator (4), as well as an infinite array of mini-batch [19] and importance sampling [23, 38] variants, and a growing list of variance-reduced variants [3]. Furthermore, there are estimators of the coordinate descent variety, including the simplest one based on (5) [26], more elaborate variants utilizing the arbitrary sampling paradigm [28], and variance reduced methods capable of handling general non-separable regularizers [10].

▷ **New general method and a single convergence theorem.** In this paper we propose a *general method*—which we call GJS—which reduces to many of the aforementioned classical and several recently developed SGD type methods in special cases. We provide a *single convergence theorem*, establishing a linear convergence rate for GJS, assuming f to be smooth and quasi strongly convex. In particular, we obtain the following methods in special cases, or their generalizations, always recovering the best-known convergence guarantees or improving upon them: SAGA [3, 27, 4], JacSketch [8], LSVRG [12, 15], SEGA [10], and ISEGA [21] (see Table 1, in which we list 17 special cases). This is the first time such a direct connection is made between many of these methods, which previously required different intuitions and dedicated analyses. Our general method, and hence also all special cases we consider, can work with a regularizer. This provides novel (although not hard) results for some methods, such as LSVRG.

▷ **First unification of SGD and RCD.** As a by-product of the generality of GJS, we obtain the *first unification of variance-reduced SGD and RCD methods*. To the best of our knowledge, there is no algorithm besides GJS, one whose complexity is captured by a single theorem, which specializes to SGD and RCD type methods at the same time.

▷ **Generalizations to arbitrary sampling.** Many specialized methods we develop are cast in a very general *arbitrary sampling* paradigm [32, 30, 28], which allows for the estimator g^k to be formed through information contained in a random subset $R^k \subseteq [n]$ (by computing $\nabla f_j(x^k)$ for $j \in R^k$) or a random subset $L^k \subseteq [d]$ (by computing $\nabla_i f(x^k)$ for $i \in L^k$), where these subsets are allowed to follow an arbitrary distribution. In particular, we extend SEGA [10], LSVRG [12, 15] or ISEGA [21] to this setup. Likewise, GJS specializes to an arbitrary sampling extension of the SGD-type method SAGA [3, 27], obtaining state-of-the-art rates. As special case, of the arbitrary sampling paradigm, we obtain *importance sampling* versions of all mentioned methods.

▷ **New methods.** GJS can be specialized to many new specific methods. To illustrate this, we construct 10 specific *new* methods in special cases, some with intriguing structure and properties (see Section 6; Table 1; and Table 2 for a summary of the rates).

▷ **Relation to JacSketch.** Our method can be seen as a vast generalization of the recently proposed Jacobian sketching method JacSketch [8] in several directions, notably by enabling *arbitrary randomized linear* (i.e., sketching) operators, extending the analysis to the proximal case, and replacing strong convexity assumption by quasi strong convexity.

▷ **Surprising connections.** Lastly, we show a surprising link between SEGA and SAGA. In particular, SAGA is a special case of SEGA; and the novel rate we obtain for SEGA recovers the tight complexity of SAGA [27, 4] (see Section R). Similarly, we recover tight rates for LSVRG using a result for SVRCD.

▷ **Limitations.** We focus on developing methods capable of enjoying a linear convergence rate with a fixed stepsize α and do not consider the non-convex setting. Although there exist several *accelerated* variance reduced algorithms [17, 1, 40, 39, 15, 16], we do not consider such methods here.

Notation. Let \mathbf{e} (resp. \mathbf{e}) be the vector of all ones in \mathbb{R}^n (resp. \mathbb{R}^d), and \mathbf{e}_j (resp. \mathbf{e}_i) be the j -th (resp. i -th) unit basis vector in \mathbb{R}^n (resp. \mathbb{R}^d). By $\|\cdot\|$ we denote the standard Euclidean norm in \mathbb{R}^d and \mathbb{R}^n . Matrices are denoted by upper-case bold letters. Given $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d \times n}$, let $\langle \mathbf{X}, \mathbf{Y} \rangle := \text{Tr}(\mathbf{X}^\top \mathbf{Y})$ and $\|\mathbf{X}\| := \langle \mathbf{X}, \mathbf{X} \rangle^{1/2}$ be the Frobenius norm. By $\mathbf{X}_{:,j}$ (resp. $\mathbf{X}_{i,:}$) we denote the j -th column (resp. i -th row) of matrix \mathbf{X} . By \mathbf{I}_n (resp. \mathbf{I}_d) we denote the $n \times n$ (resp. $d \times d$) identity matrices. Upper-

case calligraphic letters, such as $\mathcal{S}, \mathcal{U}, \mathcal{I}, \mathcal{M}, \mathcal{R}$, are used to denote (deterministic or random) linear operators mapping $\mathbb{R}^{d \times n}$ to $\mathbb{R}^{d \times n}$. Most used notation is summarized in Table 3 in Appendix C.

3 Sketching

A key object in this paper is the Jacobian matrix $\mathbf{G}(x) = [\nabla f_1(x), \dots, \nabla f_n(x)] \in \mathbb{R}^{d \times n}$. Note that

$$\nabla f(x) = \frac{1}{n} \mathbf{G}(x) \mathbf{e}. \quad (6)$$

Extending the insights from [8], one of the key observations of this work is that *random linear transformations* (sketches) of \mathbf{G} can be used to *construct* unbiased estimators of the gradient of f . For instance, $\mathbf{G}(x^k) \mathbf{e}_j$ leads to the simple SGD estimator (4), and $\frac{d}{n} \mathbf{e}_i \mathbf{e}_i^\top \mathbf{G}(x^k) \mathbf{e}$ gives the simple RCD estimator (5). We will consider more elaborate examples later on. It will be useful to embed these estimators into $\mathbb{R}^{d \times n}$. For instance, instead of $\mathbf{G}(x^k) \mathbf{e}_j$ we consider the matrix $\mathbf{G}(x^k) \mathbf{e}_j \mathbf{e}_j^\top$. Note that all columns of this matrix are zero, except for the j -th column, which is equal to $\mathbf{G}(x^k) \mathbf{e}_j$. Similarly, instead of $\frac{d}{n} \mathbf{e}_i \mathbf{e}_i^\top \mathbf{G}(x^k) \mathbf{e}$ we will consider the matrix $\frac{d}{n} \mathbf{e}_i \mathbf{e}_i^\top \mathbf{G}(x^k)$. All rows of this matrix are zero, except for the i -th row, which consists of the i th partial derivatives of functions $f_j(x^k)$ for $j \in [n]$, scaled by $\frac{d}{n}$.

Random projections. Generalizing from these examples, we consider a random linear operator (“sketch”) $\mathcal{A} : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^{d \times n}$. By \mathcal{A}^* we denote the adjoint of \mathcal{A} , i.e., linear operator satisfying $\langle \mathcal{A}\mathbf{X}, \mathbf{Y} \rangle = \langle \mathbf{X}, \mathcal{A}^*\mathbf{Y} \rangle$ for all $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d \times n}$. Given \mathcal{A} , we let $\mathcal{P}_{\mathcal{A}}$ be the (random) projection operator onto $\text{Range}(\mathcal{A}^*)$. That is,

$$\mathcal{P}_{\mathcal{A}}(\mathbf{X}) = \arg \min_{\mathbf{Y} \in \text{Range}(\mathcal{A}^*)} \|\mathbf{X} - \mathbf{Y}\| = \mathcal{A}^* (\mathcal{A} \mathcal{A}^*)^\dagger \mathcal{A} \mathbf{X},$$

where † is the Moore-Penrose pseudoinverse. The identity operator is denoted by \mathcal{I} . We say that \mathcal{A} is *identity in expectation*, or *unbiased* when $\mathbb{E}[\mathcal{A}] = \mathcal{I}$; i.e., when if $\mathbb{E}[\mathcal{A}\mathbf{X}] = \mathbf{X}$ for all $\mathbf{X} \in \mathbb{R}^{d \times n}$.

Definition 3.1. We will often consider the following⁴ sketching operators \mathcal{A} :

- (i) **Right sketch.** Let $\mathbf{R} \in \mathbb{R}^{n \times n}$ be a random matrix. Define \mathcal{A} by $\mathcal{A}\mathbf{X} = \mathbf{X}\mathbf{R}$ (“ R -sketch”). Notice that $\mathcal{A}^*\mathbf{X} = \mathbf{X}\mathbf{R}^\top$. In particular, if R is random subset of $[n]$, we can define $\mathbf{R} = \sum_{j \in R} \mathbf{e}_j \mathbf{e}_j^\top$. The resulting operator \mathcal{A} (“ R -sampling”) satisfies: $\mathcal{A} = \mathcal{A}^* = \mathcal{A}^2 = \mathcal{P}_{\mathcal{A}}$. If we let $\mathbf{p}_j := \mathbb{P}(j \in R)$, and instead define $\mathbf{R} = \sum_{j \in R} \frac{1}{\mathbf{p}_j} \mathbf{e}_j \mathbf{e}_j^\top$, then $\mathbb{E}[\mathbf{R}] = \mathbf{I}_n$ and hence \mathcal{A} is unbiased.
- (ii) **Left sketch.** Let $\mathbf{L} \in \mathbb{R}^{d \times d}$ be a random matrix. Define \mathcal{A} by $\mathcal{A}\mathbf{X} = \mathbf{L}\mathbf{X}$ (“ L -sketch”). Notice that $\mathcal{A}^*\mathbf{X} = \mathbf{L}^\top \mathbf{X}$. In particular, if L is random subset of $[d]$, we can define $\mathbf{L} = \sum_{i \in L} \mathbf{e}_i \mathbf{e}_i^\top$. The resulting operator \mathcal{A} (“ L -sampling”) satisfies: $\mathcal{A} = \mathcal{A}^* = \mathcal{A}^2 = \mathcal{P}_{\mathcal{A}}$. If we let $\mathbf{p}_i := \mathbb{P}(i \in L)$, and instead define $\mathbf{L} = \sum_{i \in L} \frac{1}{\mathbf{p}_i} \mathbf{e}_i \mathbf{e}_i^\top$, then $\mathbb{E}[\mathbf{L}] = \mathbf{I}_d$ and hence \mathcal{A} is unbiased.
- (iii) **Scaling/Bernoulli.** Let ξ be a Bernoulli random variable, i.e., $\xi = 1$ with probability ρ and $\xi = 0$ with probability $1 - \rho$, where $\rho \in [0, 1]$. Define \mathcal{A} by $\mathcal{A}\mathbf{X} = \xi \mathbf{X}$ (“scaling”). Then $\mathcal{A} = \mathcal{A}^* = \mathcal{A}^2 = \mathcal{P}_{\mathcal{A}}$. If we instead define $\mathcal{A}\mathbf{X} = \frac{1}{\rho} \xi \mathbf{X}$, then \mathcal{A} is unbiased.
- (iv) **LR sketch.** All the above operators can be combined. In particular, we can define $\mathcal{A}\mathbf{X} = \xi \mathbf{L} \mathbf{X} \mathbf{R}$. All of the above arise as special cases of this: (i) arises for $\xi \equiv 1$ and $\mathbf{L} \equiv \mathbf{I}_d$, (ii) for $\xi \equiv 1$ and $\mathbf{R} \equiv \mathbf{I}_n$, and (iii) for $\mathbf{L} \equiv \mathbf{I}_d$ and $\mathbf{R} \equiv \mathbf{I}_n$.

4 Generalized Jacobian Sketching (GJS)

We are now ready to describe our method (formalized as Algorithm 1). Let \mathcal{S} be a random linear operator (e.g., right sketch, left sketch, or scaling) such that $\mathcal{S} = \mathcal{P}_{\mathcal{S}}$ and let \mathcal{U} be an unbiased

⁴The algorithm we develop is, however, not limited to such sketches.

Algorithm 1 Generalized JacSketch (GJS)

- 1: **Parameters:** Stepsize $\alpha > 0$, random projector \mathcal{S} and unbiased sketch \mathcal{U}
 - 2: **Initialization:** Choose solution estimate $x^0 \in \mathbb{R}^d$ and Jacobian estimate $\mathbf{J}^0 \in \mathbb{R}^{d \times n}$
 - 3: **for** $k = 0, 1, \dots$ **do**
 - 4: Sample realizations of \mathcal{S} and \mathcal{U} , and perform sketches $\mathcal{S}\mathbf{G}(x^k)$ and $\mathcal{U}\mathbf{G}(x^k)$
 - 5: $\mathbf{J}^{k+1} = \mathbf{J}^k - \mathcal{S}(\mathbf{J}^k - \mathbf{G}(x^k))$ update the Jacobian estimate via (9)
 - 6: $g^k = \frac{1}{n}\mathbf{J}^k e + \frac{1}{n}\mathcal{U}(\mathbf{G}(x^k) - \mathbf{J}^k)e$ construct the gradient estimator via (7)
 - 7: $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$ perform the proximal SGD step (3)
 - 8: **end for**
-

operator. We propose to construct the gradient estimator as

$$g^k = \frac{1}{n}\mathbf{J}^k e + \frac{1}{n}\mathcal{U}(\mathbf{G}(x^k) - \mathbf{J}^k)e, \quad (7)$$

where the matrices $\mathbf{J}^k \in \mathbb{R}^{d \times n}$ are constructed iteratively. Note that, taking expectation in \mathcal{U} , we get

$$\mathbb{E}[g^k] \stackrel{(7)}{=} \frac{1}{n}\mathbf{J}^k e + \frac{1}{n}(\mathbf{G}(x^k) - \mathbf{J}^k)e = \frac{1}{n}\mathbf{G}(x^k)e \stackrel{(6)}{=} \nabla f(x^k), \quad (8)$$

and hence g^k is indeed unbiased. We will construct \mathbf{J}^k so that $\mathbf{J}^k \rightarrow \mathbf{G}(x^*)$. By doing so, the variance of g^k decreases throughout the iterations, completely vanishing at x^* . The sequence $\{\mathbf{J}^k\}$ is updated as follows:

$$\mathbf{J}^{k+1} = \arg \min_{\mathbf{J}} \{\|\mathbf{J} - \mathbf{J}^k\| : \mathcal{S}\mathbf{J} = \mathcal{S}\mathbf{G}(x^k)\} = \mathbf{J}^k - \mathcal{S}(\mathbf{J}^k - \mathbf{G}(x^k)). \quad (9)$$

That is, we sketch the Jacobian $\mathbf{G}(x^k)$, obtaining the sketch $\mathcal{S}\mathbf{G}(x^k)$, and seek to use this information to construct a new matrix \mathbf{J}^{k+1} which is consistent with this sketch, and as close to \mathbf{J}^k as possible. The intuition here is as follows: if we repeated the sketch-and-project process (9) for fixed x^k , the matrices \mathbf{J}^k would converge to $\mathbf{G}(x^k)$, at a linear rate [6, 7]. This process can be seen as SGD applied to a certain quadratic stochastic optimization problem [34, 8]. Instead, we take just one step of this iterative process, change x^k , and repeat. Note that the unbiased sketch \mathcal{U} in (7) also claims access to $\mathbf{G}(x^k)$. Specific variants of GJS are obtained by choosing specific operators \mathcal{S} and \mathcal{U} (see Section 6).

5 Theory

We now describe the main result of this paper, which depends on a relaxed strong convexity assumption and a more precise smoothness assumption on f .

Assumption 5.1. *Problem (1) has a unique minimizer x^* , and f is σ -quasi strongly convex, i.e.,*

$$f(x^*) \geq f(y) + \langle \nabla f(y), x^* - y \rangle + \frac{\sigma}{2} \|y - x^*\|^2, \quad \forall y \in \mathbb{R}^d, \quad (10)$$

Functions f_j are convex and \mathbf{M}_j -smooth for some $\mathbf{M}_j \succeq 0$, i.e.,

$$f_j(y) + \langle \nabla f_j(y), x - y \rangle \leq f_j(x) \leq f_j(y) + \langle \nabla f_j(y), x - y \rangle + \frac{1}{2} \|y - x\|_{\mathbf{M}_j}^2, \quad \forall x, y \in \mathbb{R}^d. \quad (11)$$

Assumption 11 generalizes classical L -smoothness, which is obtained in the special case $\mathbf{M}_j = L\mathbf{I}_d$. The usefulness of this assumption comes from i) the fact that ERM problems typically satisfy (11) in a non-trivial way [29, 5], ii) our method is able to utilize the full information contained in these matrices for further acceleration (via increased stepsizes). Given matrices $\{\mathbf{M}_j\}$ from Assumption 5.1, let \mathcal{M} be the linear operator defined via $(\mathcal{M}\mathbf{X})_{:j} = \mathbf{M}_j\mathbf{X}_{:j}$ for $j \in [n]$. It is easy to check that this operator is self-adjoint and positive semi-definite, and that its square root is given by $(\mathcal{M}^{1/2}\mathbf{X})_{:j} = \mathbf{M}_j^{1/2}\mathbf{X}_{:j}$. The pseudoinverse \mathcal{M}^\dagger of this operator plays an important role in our main result.

Theorem 5.2. *Let Assumption 5.1 hold. Let \mathcal{B} be any linear operator commuting with \mathcal{S} , and assume $\mathcal{M}^{\dagger 1/2}$ commutes with \mathcal{S} . Let \mathcal{R} be any linear operator for which $\mathcal{R}(\mathbf{J}^k) = \mathcal{R}(\mathbf{G}(x^*))$ for every $k \geq 0$. Define the Lyapunov function*

$$\Psi^k := \|x^k - x^*\|^2 + \alpha \|\mathcal{B}(\mathbf{J}^k - \mathbf{G}(x^*))\|^2, \quad (12)$$

where $\{x^k\}$ and $\{\mathbf{J}^k\}$ are the random iterates produced by Algorithm 1 with stepsize $\alpha > 0$. Suppose that α and \mathcal{B} are chosen so that

$$\frac{2\alpha}{n^2} \mathbb{E} [\|\mathcal{U}\mathbf{X}\mathbf{e}\|^2] + \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} \mathbf{X} \right\|^2 \leq (1 - \alpha\sigma) \left\| \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} \mathbf{X} \right\|^2 \quad (13)$$

whenever $\mathbf{X} \in \text{Range}(\mathcal{R})^\perp$ and

$$\frac{2\alpha}{n^2} \mathbb{E} [\|\mathcal{U}\mathbf{X}\mathbf{e}\|^2] + \left\| (\mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} \mathbf{X} \right\|^2 \leq \frac{1}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} \mathbf{X} \right\|^2. \quad (14)$$

for all $\mathbf{X} \in \mathbb{R}^{d \times n}$. Then for all $k \geq 0$, we have $\mathbb{E}[\Psi^k] \leq (1 - \alpha\sigma)^k \Psi^0$.

The above theorem is very general as it applies to essentially arbitrary random linear operators \mathcal{S} and \mathcal{U} . It postulates a linear convergence rate of a Lyapunov function composed of two terms: distance of x^k from x^* , and weighted distance of the Jacobian \mathbf{J}^k from $\mathbf{G}(x^*)$. Hence, we obtain convergence of both the iterates and the Jacobian to x^* and $\mathbf{G}(x^*)$, respectively. Inequalities (13) and (14) are mainly assumptions on stepsize α , and are used to define suitable weight operator \mathcal{B} . See Lemma E.1 for a general statement on when these inequalities are satisfied. However, we give concrete and simple answers in all special cases of GJS in the appendix. For a summary of how the operator \mathcal{B} is chosen in special cases, and the particular complexity results derived from this theorem, we refer to Table 2.

Remark 5.3. *We use the trivial choice $\mathcal{R} \equiv 0$ in almost all special cases. With this choice of \mathcal{R} , the condition $\mathcal{R}(\mathbf{J}^k) = \mathcal{R}(\mathbf{G}(x^*))$ is automatically satisfied, and inequality (14) is requested to hold for all matrices $\mathbf{X} \in \mathbb{R}^{d \times n}$. However, a non-trivial choice of \mathcal{R} is sometimes useful; e.g., in the analysis of a subspace variant of SEGA [10]. Further, the results of Theorem 5.2 can be generalized from a quasi strong convexity to a strong growth condition [14] on f (see Section S). While interesting, these are not the key results of this work and we therefore suppress them to the appendix.*

6 Special Cases

As outlined in the introduction, GJS (Algorithm 1) is a surprisingly versatile method. In Table 1 we list 7 existing methods (in some cases, generalizations of existing methods), and construct also 10 new variance reduced methods. We also provide a summary of all specialized iteration complexity results, and a guide to the corollaries which state them (see Table 2 in the appendix).

▷ **SGD-shift.** In order to illustrate why variance reduction is needed in the first place, let us start by describing one of the new methods—SGD-shift (Algorithm 7)—which happens to be particularly suitable to shed light on this issue. In SGD-shift we assume that the Jacobian at optimum, $\mathbf{G}(x^*)$, is known. While this is clearly an unrealistic assumption, let us see where it leads us. If this is the case, we can choose $\mathbf{J}^0 = \mathbf{G}(x^*)$, and let $\mathcal{S} \equiv 0$. This implies that $\mathbf{J}^k = \mathbf{J}^0$ for all k . We then choose \mathcal{U} to be the right unbiased sampling operator, i.e., $\mathcal{U}\mathbf{X} = \mathbf{X} \sum_{j \in R} \frac{1}{p_j} \mathbf{e}_j \mathbf{e}_j^\top$, which gives

$$g^k = \frac{1}{n} \sum_{j=1}^n \nabla f_j(x^*) + \sum_{j \in R^k} \frac{1}{np_j} (\nabla f_j(x^k) - \nabla f_j(x^*)).$$

This method does not need to learn the Jacobian at x^* as it is known, and instead moves in a direction of average gradient at the optimum, perturbed by a random estimator of the direction $\nabla f(x^k) - \nabla f(x^*)$ formed via sub-sampling $j \in R^k \subseteq [n]$. What is special about this perturbation?

Choice of random operators \mathcal{S} and \mathcal{U} defining Algorithm 1		Algorithm			
$\mathcal{S}\mathbf{X}$	$\mathcal{U}\mathbf{X}$	#	Name	Comment	Section
$\mathbf{X}e_j e_j^\top$ w.p. $p_j = \frac{1}{n}$	$\mathbf{X}n e_j e_j^\top$ w.p. $p_j = \frac{1}{n}$	2	SAGA	basic variant of SAGA [3]	G.1
$\mathbf{X} \sum_{j \in R} e_j e_j^\top$ w.p. p_R	$\mathbf{X} \sum_{j \in R} \frac{1}{p_j} e_j e_j^\top$ w.p. p_R	3	SAGA	SAGA with AS [27]	G.2
$e_i e_i^\top \mathbf{X}$ w.p. $p_i = \frac{1}{d}$	$d e_i e_i^\top \mathbf{X}$ w.p. $p_i = \frac{1}{d}$	4	SEGA	basic variant of SEGA [10]	H.1
$\sum_{i \in L} e_i e_i^\top \mathbf{X}$ w.p. p_L	$\sum_{i \in L} \frac{1}{p_i} e_i e_i^\top \mathbf{X}$ w.p. p_L	5	SEGA	SEGA [10] with AS and prox	H.2
$= \begin{cases} 0 & \text{w.p. } 1 - \rho \\ \mathbf{X} & \text{w.p. } \rho \end{cases}$	$\sum_{i \in L} \frac{1}{p_i} e_i e_i^\top \mathbf{X}$ w.p. p_L	6	SVRCD	NEW	H.3
0	$\mathbf{X} \sum_{j \in R} \frac{1}{p_j} e_j e_j^\top$ w.p. p_R	7	SGD-shift	NEW	I
$= \begin{cases} 0 & \text{w.p. } 1 - \rho \\ \mathbf{X} & \text{w.p. } \rho \end{cases}$	$\mathbf{X} \sum_{j \in R} \frac{1}{p_j} e_j e_j^\top$ w.p. p_R	8	LSVRG	LSVRG [15] with AS and prox	J
$= \begin{cases} 0 & \text{w.p. } 1 - \rho \\ \mathbf{X} & \text{w.p. } \rho \end{cases}$	$= \begin{cases} 0 & \text{w.p. } 1 - \delta \\ \frac{1}{\delta} \mathbf{X} & \text{w.p. } \delta \end{cases}$	9	B2	NEW	K.1
$\mathbf{X} \sum_{j \in R} e_j e_j^\top$ w.p. p_R	$= \begin{cases} 0 & \text{w.p. } 1 - \delta \\ \frac{1}{\delta} \mathbf{X} & \text{w.p. } \delta \end{cases}$	10	LSVRG-inv	NEW	K.2
$\sum_{i \in L} e_i e_i^\top \mathbf{X}$ w.p. p_L	$= \begin{cases} 0 & \text{w.p. } 1 - \delta \\ \frac{1}{\delta} \mathbf{X} & \text{w.p. } \delta \end{cases}$	11	SVRCD-inv	NEW	K.3
$\mathbf{X} \sum_{j \in R} e_j e_j^\top$ w.p. p_R	$\sum_{i \in L} \frac{1}{p_i} e_i e_i^\top \mathbf{X}$ w.p. p_L	12	RL	NEW	L.1
$\sum_{i \in L} e_i e_i^\top \mathbf{X}$ w.p. p_L	$\mathbf{X} \sum_{j \in R} \frac{1}{p_j} e_j e_j^\top$ w.p. p_R	13	LR	NEW	L.2
$\mathbf{I}_L, \mathbf{X}\mathbf{I}_R$ w.p. $p_L p_R$	$\mathbf{I}_L: \left((p^{-1} (p^{-1})^\top) \circ \mathbf{X} \right) \mathbf{I}_R$ w.p. $p_L p_R$	14	SAEGA	NEW	M.1
$= \begin{cases} 0 & \text{w.p. } 1 - \rho \\ \mathbf{X} & \text{w.p. } \rho \end{cases}$	$\mathbf{I}_L: \left((p^{-1} (p^{-1})^\top) \circ \mathbf{X} \right) \mathbf{I}_R$ w.p. $p_L p_R$	15	SVRCDG	NEW	M.2
$\sum_{t=1}^T \mathbf{I}_{L_t}: \mathbf{X}: \mathbf{N}_t \mathbf{I}_{R_t}$	$\sum_{t=1}^T \left((p^t)^{-1} (p^t)^{-1\top} \right) \circ (\mathbf{I}_{L_t}: \mathbf{X}: \mathbf{N}_t \mathbf{I}_{R_t})$	16	ISAEGA	NEW (reminiscent of [21])	M.3
$\sum_{t=1}^T \mathbf{I}_{L_t}: \mathbf{X}: \mathbf{N}_t$	$\sum_{t=1}^T \left((p^t)^{-1} \mathbf{e}^\top \right) \circ (\mathbf{I}_{L_t}: \mathbf{X}: \mathbf{N}_t)$	17	ISEGA	ISEGA [21] with AS	M.3
$\mathbf{X}\mathbf{R}$	$\mathbf{X}\mathbf{R}\mathbf{E}[\mathbf{R}]^{-1}$	18	JS	JacSketch [8] with AS and prox	N

Table 1: Selected special cases of GJS (Alg. 1) arising by choosing operators \mathcal{S} and \mathcal{U} in particular ways. R is a random subset of $[n]$, L is a random subset of $[d]$, $p_i = \mathbb{P}(i \in L)$, $p_j = \mathbb{P}(j \in R)$.

As the method converges, $x^k \rightarrow x^*$ and the perturbations converge to zero, for *any* realization of the random set $R^k \subseteq [n]$. So, gradient estimation stabilizes, we get $g^k \rightarrow \nabla f(x^*)$, and hence the variance of g^k converges to zero. In view of Corollary I.1 of our main result (Theorem 5.2), the iteration complexity of SGD-shift is $\max_j \frac{v_j}{\sigma n p_j} \log \frac{1}{\epsilon}$, where σ is the quasi strong convexity parameter of f , and the smoothness constants v_j are defined in Section I.

Since knowing $\mathbf{G}(x^*)$ is unrealistic, GJS is instead *learning* these perturbations on the fly. Different variants of GJS do this differently, but ultimately all attempt to learn the gradients $\nabla f_j(x^*)$ and use this information to stabilize the gradient estimation. Due to space restrictions, we do not describe all remaining 9 new methods in the main body of the paper, let alone the all 17 methods. We will briefly outline 2 more (not necessarily the most interesting) new methods:

▷ SVRCD. This method belongs to the RCD variety, and constructs the gradient estimator via the rule

$$g^k = h^k + \sum_{i \in L^k} \frac{1}{p_i} (\nabla_i f(x^k) - h_i^k) e_i,$$

where $L^k \subseteq [d]$ is sampled afresh in each iteration. The auxiliary vector h^k is updated using a simple biased coin flip: $h^{k+1} = h^k$ with probability $1 - \rho$, and $h^{k+1} = \nabla f(x^k)$ with probability ρ . So, a full pass over all coordinates is made in each iteration with probability ρ , and a partial derivatives $\nabla_i f(x^k)$ for $i \in L^k$ are computed in each iteration. This method has a similar structure to LSVRG, which instead sub-sampling coordinates sub-samples functions f_j for $j \in R^k$ (see Table 1). The iteration complexity of this method is $\left(\frac{1}{\rho} + \max_i \frac{1}{p_i} \frac{4m_i}{\sigma} \right) \log \frac{1}{\epsilon}$, where m_i is a smoothness parameter of f associated with coordinate i (see Table 2 and Corollary H.4).

▷ **ISAEGA**. In [21], a strategy of running RCD on top of a parallel implementation of optimization algorithms such as GD, SGD or SAGA was proposed. Surprisingly, it was shown that the runtime of the overall algorithm is unaffected whether one computes and communicates *all entries* of the stochastic gradient on each worker, or only a *fraction* of all entries of size inversely proportional to the number of all workers. However, ISAGA [21] (distributed SAGA with RCD on top of it), as proposed, requires the gradients with respect to the data owned by a given machine to be zero at the optimum. On the other hand, ISEGA [21] does not have the issue, but it requires a computation of the exact partial derivatives on each machine and thus is expensive. As a special case of GJS we propose **ISAEGA** – a method which cherry-picks the best properties from both ISAGA (allowing for stochastic partial derivatives) and ISEGA (not requiring zero gradients at the optimum). Further, we present the method in the arbitrary sampling paradigm. See Section M.3 for more details.

7 Experiments

We perform extensive numerical testing for various special cases of Algorithm 1. Due to space limitations, we only give a quick taste using a single experiment here. The complete numerical evaluation is presented in Section D of the supplementary.

In particular, in Section D.1 we demonstrate that SEGA with importance sampling outperforms both basic SEGA and proximal gradient descent, often significantly. Next, Section D.2 demonstrates that, as predicted by theory, convergence speed of SVRG is influenced by the choice of ρ only weakly. Further, in Section D.3 we demonstrate the claimed linear parallel scaling of ISAEGA (in the sense of [21]). Lastly, in Section D.4 we demonstrate the superiority of LSVRG with importance sampling (a new variant of LSVRG obtained here) to plain LSVRG, plain SAGA and SAGA with importance sampling. We only outline the last experiment here; a complete description is given in Section D.4.

We consider a logistic regression problem on LibSVM [2]. In order to conduct fair testing, we only compare methods where the expected minibatch size is fixed. We set $\mathbb{E}[|R|] = \tau$ and compare LSVRG with $\rho = \frac{1}{2n}$, and SAGA with importance sampling (imp) and uniform sampling (unif). The results are presented in Figure 1.

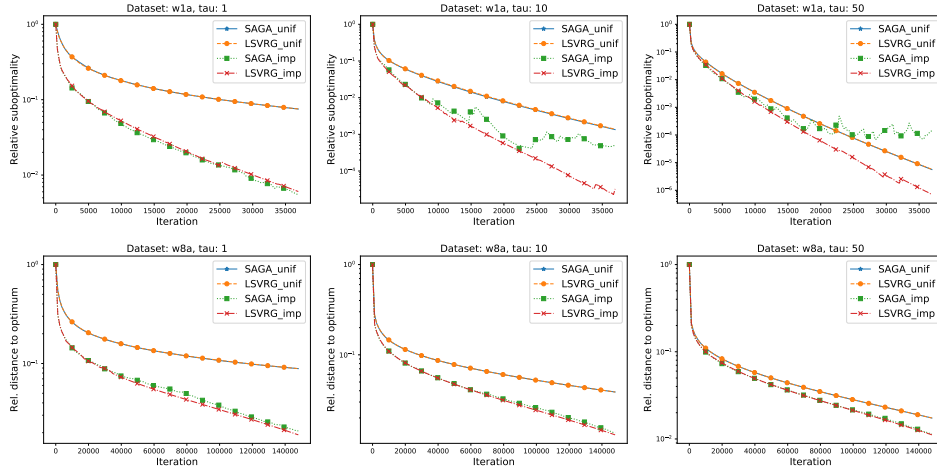


Figure 1: Comparison of LSVRG & SAGA with importance and uniform sampling.

In all cases, LSVRG with importance sampling was the fastest method, while uniform LSVRG and SAGA performed almost identically. The gain from importance sampling is noticeable for small τ . For larger τ , importance sampling is less significantly superior. Note that this behavior is predicted by theory.

However, our experiments indicate the superiority of LSVRG to SAGA in the importance sampling setup. In particular, proposed stepsize γ (see Section D.4) is often too large for SAGA. Note that the optimal stepsize requires the prior knowledge of the quasi strong convexity constant⁵ σ , which is in our case unknown (see the importance serial sampling proposed in [8]). While one can still estimate it as the L_2 regularization constant, this would be a weak estimate and yield suboptimal performance.

References

- [1] Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1200–1205. ACM, 2017.
- [2] Chih-Chung Chang and Chih-Jen Lin. LibSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.
- [3] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.
- [4] Nidham Gazagnadou, Robert M Gower, and Joseph Salmon. Optimal mini-batch and step sizes for SAGA. *arXiv preprint arXiv:1902.00071*, 2019.
- [5] Robert M Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. SGD: General analysis and improved rates. *arXiv preprint arXiv:1901.09401*, 2019.
- [6] Robert M Gower and Peter Richtárik. Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1660–1690, 2015.
- [7] Robert M Gower and Peter Richtárik. Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1380–1409, 2017.
- [8] Robert M Gower, Peter Richtárik, and Francis Bach. Stochastic quasi-gradient methods: Variance reduction via Jacobian sketching. *arXiv preprint arXiv:1805.02632*, 2018.
- [9] Robert M Gower, Peter Richtárik, and Francis Bach. Stochastic quasi-gradient methods: Variance reduction via Jacobian sketching. *arXiv preprint arXiv:1805.02632*, 2018.
- [10] Filip Hanzely, Konstantin Mishchenko, and Peter Richtárik. SEGA: Variance reduction via gradient sketching. In *Advances in Neural Information Processing Systems*, pages 2082–2093, 2018.
- [11] Filip Hanzely and Peter Richtárik. Accelerated coordinate descent with arbitrary sampling and best rates for minibatches. In *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 304–312. PMLR, 16–18 Apr 2019.
- [12] Thomas Hofmann, Aurelien Lucchi, Simon Lacoste-Julien, and Brian McWilliams. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems*, pages 2305–2313, 2015.
- [13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

⁵Or more generally, knowledge of the strong growth constant is required; see Section S.

- [14] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Lojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.
- [15] 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. *arXiv preprint arXiv:1901.08689*, 2019.
- [16] Andrei Kulunchakov and Julien Mairal. Estimate sequences for stochastic composite optimization: Variance reduction, acceleration, and robustness to noise. *arXiv preprint arXiv:1901.08788*, 2019.
- [17] Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *Mathematical programming*, 171(1-2):167–215, 2018.
- [18] Lihua Lei and Michael Jordan. Less than a single pass: Stochastically controlled stochastic gradient. In *Artificial Intelligence and Statistics*, pages 148–156, 2017.
- [19] Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J Smola. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’14, pages 661–670, New York, NY, USA, 2014. ACM.
- [20] Julien Mairal. Optimization with first-order surrogate functions. In *International Conference on Machine Learning*, pages 783–791, 2013.
- [21] Konstantin Mishchenko, Filip Hanzely, and Peter Richtárik. 99% of parallel optimization is inevitably a waste of time. *arXiv preprint arXiv:1901.09437*, 2019.
- [22] Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011.
- [23] Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Mathematical Programming*, 155(1–2):549–573, 2015.
- [24] Deanna Needell and Rachel Ward. Batched stochastic gradient descent with weighted sampling. In *International Conference Approximation Theory*, pages 279–306. Springer, 2016.
- [25] Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. In *Advances in Neural Information Processing Systems*, pages 1017–1025, 2014.
- [26] Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.
- [27] Xun Qian, Zheng Qu, and Peter Richtárik. SAGA with arbitrary sampling. *arXiv preprint arXiv:1901.08669*, 2019.
- [28] Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling I: Algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, 2016.
- [29] Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling II: Expected separable overapproximation. *Optimization Methods and Software*, 31(5):858–884, 2016.

- [30] Zheng Qu, Peter Richtárik, and Tong Zhang. Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *Advances in Neural Information Processing Systems*, pages 865–873, 2015.
- [31] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(2):1–38, 2014.
- [32] Peter Richtárik and Martin Takáč. On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters*, 10(6):1233–1243, 2016.
- [33] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1):433–484, 2016.
- [34] Peter Richtárik and Martin Takáč. Stochastic reformulations of linear systems: algorithms and convergence theory. *arXiv:1706.01108*, 2017.
- [35] Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems*, pages 2663–2671, 2012.
- [36] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *Journal of Machine Learning Research*, 14(1):567–599, 2013.
- [37] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.
- [38] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *Proceedings of the 32nd International Conference on Machine Learning*, PMLR, volume 37, pages 1–9, 2015.
- [39] Kaiwen Zhou, Qinghua Ding, Fanhua Shang, James Cheng, Danli Li, and Zhi-Quan Luo. Direct acceleration of SAGA using sampled negative momentum. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 1602–1610. PMLR, 16–18 Apr 2019.
- [40] Kaiwen Zhou, Fanhua Shang, and James Cheng. A simple stochastic variance reduced algorithm with fast convergence rates. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5980–5989. PMLR, 10–15 Jul 2018.

Supplementary Material for: One Method to Rule Them All: Variance Reduction for Data, Parameters and Many New Methods

A Table of Contents

For easier navigation through the paper and the supplementary material, we include a table of contents.

Contents

1	Introduction	1
2	Contributions	2
3	Sketching	4
4	Generalized Jacobian Sketching (GJS)	4
5	Theory	5
6	Special Cases	6
7	Experiments	8
A	Table of Contents	13
B	Summary of Complexity Results	16
C	Table of Frequently Used Notation	17
D	Additional Experiments	19
D.1	SEGA and SVRCD with importance sampling	19
D.2	SVRCD: Effect of ρ	19
D.3	ISAEGA	19
D.4	LSVRG with importance sampling	22
E	Several Lemmas	25
E.1	Existence lemma	25
E.2	Smoothness lemmas	25
E.3	Projection lemma	27
E.4	Decomposition lemma	27

F	Proof of Theorem 5.2	28
G	Special Cases: SAGA-like Methods	30
G.1	Basic variant of SAGA [3]	30
G.2	SAGA with arbitrary sampling	30
H	Special Cases: SEGA-like Methods	31
H.1	Basic variant of SEGA [10]	31
H.2	SEGA with arbitrary sampling	31
H.3	SVRCD with arbitrary sampling	32
I	Special Cases: SGD-shift	33
J	Special Cases: Loopless SVRG with Arbitrary Sampling (LSVRG)	34
K	Special Cases: Methods with Bernoulli \mathcal{U}	34
K.1	B2 (Bernoulli \mathcal{S})	34
K.2	LSVRG-inv (Right \mathcal{S})	35
K.3	SVRCD-inv (Left \mathcal{S})	36
L	Special Cases: Combination of Left and Right Sketches	36
L.1	RL (right sampling \mathcal{S} , left unbiased sampling \mathcal{U})	36
L.2	LR (left sampling \mathcal{S} , right unbiased sampling \mathcal{U})	36
M	Special Cases: Joint Left and Right Sketches	37
M.1	SAEGA	37
M.2	SVRCDG	38
M.3	ISAEGA (with distributed data)	39
N	Special Cases: JacSketch	40
O	Special Cases: Proofs	42
O.1	SAGA methods: Proofs	42
O.1.1	Setup for Corollary G.1	42
O.1.2	Setup for Corollary G.3	42
O.2	SEGA methods: Proofs	43
O.2.1	Setup for Corollary H.1	43
O.2.2	Setup for Corollary H.2	43
O.2.3	Setup for Corollary H.4	43

O.3	Setup for Corollary I.1	44
O.4	Setup for Corollary J.1	44
O.5	Methods with Bernoulli \mathcal{U} : Proofs	45
O.5.1	Setup for Corollary K.1	45
O.5.2	Setup for Corollary K.3	45
O.5.3	Setup for Corollary K.4	45
O.6	Combination of left and right sketches (in different operators): Proofs	46
O.6.1	Setup for Corollary L.1	46
O.6.2	Setup for Corollary L.2	46
O.7	Joint Sketches: Proofs	47
O.7.1	Setup for Corollary M.1	47
O.7.2	Setup for Corollary M.2	47
O.7.3	Setup for Corollary M.3	48
O.8	Setup for Corollary N.1	49
P	Tighter Rates for Left \mathcal{U} by Exploiting Prox	50
Q	Even faster rates for SEGA	52
Q.1	Setup for Corollary Q.1	52
R	SAGA with Arbitrary Sampling via SEGA	53
R.1	Proof of Lemma R.1	53
S	Convergence Under Strong Growth Condition	54
S.1	Technical proposition and lemma	54
S.2	Theorem	54

B Summary of Complexity Results

We provide a comprehensive table for faster navigation through special cases and their iteration complexities. In particular, for each special case of GJS, we provide the leading complexity term (i.e., a $\log \frac{1}{\varepsilon}$ factor is omitted in all results) and a reference to the corresponding corollary where this result is established. We also indicate how the operator \mathcal{B} appearing in the Lyapunov function is picked (this is not needed to run the method; it is only used in the analysis). All details can be found later in the Appendix.

Algorithm		Theory		
#	Name	Cor. of Thm 5.2	$\mathcal{B}\mathbf{X}$	Leading complexity term (i.e., $\log \frac{1}{\varepsilon}$ factor omitted)
2	SAGA	Corollary G.1	$\beta\mathbf{X}$	$n + \frac{4m}{\sigma}$
3	SAGA	Corollary G.3	$\mathbf{XD}(b)$	$\max_j \left(\frac{1}{p_j} + \frac{1}{p_j} \frac{4v_j}{\sigma n} \right)$
4	SEGA	Corollary H.1	$\beta\mathbf{X}$	$d + d \frac{4m}{\sigma}$
5	SEGA	Corollary H.2	$\mathbf{D}(b)\mathbf{X}$	$\max_i \left(\frac{1}{p_i} + \frac{1}{p_i} \frac{4m_i}{\sigma} \right)$
6	SVRCD	Corollary H.4	$\mathbf{D}(b)\mathbf{X}$	$\frac{1}{\rho} + \max_i \frac{1}{p_i} \frac{4m_i}{\sigma}$
7	SGD-shift	Corollary I.1	$\mathbf{XD}(b)$	$\max_j \frac{1}{p_j} \frac{v_j}{\sigma n}$
8	LSVRG	Corollary J.1	$\mathbf{XD}(b)$	$\frac{1}{\rho} + \max_j \frac{1}{p_j} \frac{4v_j}{\sigma n}$
9	B2	Corollary K.1	$\beta\mathbf{X}$	$\frac{1}{\rho} + \frac{1}{\delta} \frac{4m}{\sigma}$
10	LSVRG-inv	Corollary K.3	$\mathbf{XD}(b)$	$\max_j \frac{1}{p_j} + \frac{1}{\delta} \frac{4m}{\sigma}$
11	SVRCD-inv	Corollary K.4	$\mathbf{D}(b)\mathbf{X}$	$\max_i \frac{1}{p_i} + \frac{1}{\delta} \frac{4m}{\sigma}$
12	RL	Corollary L.1	$\mathbf{XD}(b)$	$\max_{i,j} \left(\frac{1}{p_j} + \frac{1}{p_i} \frac{4m_i^j}{\sigma} \right)$
13	LR	Corollary L.2	$\mathbf{D}(b)\mathbf{X}$	$\max_{i,j} \left(\frac{1}{p_i} + \frac{1}{p_j} \frac{4v_j}{\sigma} \right)$
14	SAEGA	Corollary M.1	$\mathbf{B} \circ \mathbf{X}$	$\max_{i,j} \left(\frac{1}{p_i q_j} + \frac{1}{p_i q_j} \frac{4m_i^j}{\sigma n} \right)$
15	SVRCDG	Corollary M.2	$\beta \circ \mathbf{X}$	$\frac{1}{\rho} + \max_{i,j} \frac{1}{p_i q_j} \frac{4m_i^j}{\sigma n}$
16	ISAEGA	Corollary M.3	$\mathbf{B} \circ \mathbf{X}$	$\max_{j \in N_t, i, t} \left(\frac{1}{p_i^t q_j^t} + \left(1 + \frac{1}{n p_i^t q_j^t} \right) \frac{4m_i^j}{\sigma} \right)$
17	ISEGA	Corollary M.5	$\mathbf{B} \circ \mathbf{X}$	$\max_{j \in N_t, i, t} \left(\frac{1}{p_i^t N_t } + \left(1 + \frac{1}{n p_i^t N_t } \right) \frac{4m_i^j}{\sigma} \right)$
18	JS	Corollary N.1	$\beta\mathbf{XB}$	$\frac{4n^{-1}\eta\sigma^{-1}\lambda_{\max}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}]\mathbf{B}) + \lambda_{\max}(\mathbf{B}^\top \mathbf{B})}{\lambda_{\min}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}]\mathbf{B})}$

Table 2: Iteration complexity of selected special cases of GJS (Algorithm 1). Whenever m appears in a result, we assume that $\mathbf{M}_j = m\mathbf{I}_d$ for all j (i.e., f_j is m -smooth). Whenever m_i appears in a result, we assume that f is \mathbf{M} -smooth with $\mathbf{M} = \mathbf{D}(m_1, \dots, m_d)$. Whenever m_i^j appears in a result, we assume that $\mathbf{M}_j = \mathbf{D}(m_1^j, \dots, m_d^j)$. Quantities p_i for $i \in [d]$, p_j for $j \in [n]$, ρ and δ are probabilities defining the algorithms.

C Table of Frequently Used Notation

Due to the generality of Algorithm 1, which gives rise to a large number existing and new methods in particular cases, we appreciate that this paper is rather notation-heavy – and this is still the case after us having spent a considerable amount of time simplifying and optimizing the notation. In an attempt to make the paper more easy to read, here we include a table of the most frequently used notation. We recommend the reader to consult this table while studying our results.

Table 3: Frequently used notation.

Functions	
$f_j : \mathbb{R}^d \rightarrow \mathbb{R}$	a differentiable convex function
$f : \mathbb{R}^d \rightarrow \mathbb{R}$	$f(x) = \frac{1}{n} \sum_{j=1}^n f_j(x)$
$\nabla f : \mathbb{R}^d \rightarrow \mathbb{R}^d$	gradient of f
$\nabla_i f : \mathbb{R}^d \rightarrow \mathbb{R}$	i -th partial derivative of f
ψ	regularizer $\mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ (a proper closed convex function)
$x^* \in \mathbb{R}^d$	unique minimizer of $f + \psi$
σ	(a positive) quasi strong convexity constant of f (see (10))
Sets	
$[n]$	the set $\{1, 2, \dots, n\}$
$[d]$	the set $\{1, 2, \dots, d\}$
R	random subset (“sampling”) of $[n]$
R^k	random subset (“sampling”) of $[n]$ drawn at iteration k
L	a random subset (“sampling”) of $[d]$
L^k	random subset (“sampling”) of $[d]$ drawn at iteration k
p_j	probability that $j \in R$
p_i	probability that $i \in L$
Spaces \mathbb{R}^n and \mathbb{R}^d	
$\mathbf{e} \in \mathbb{R}^n$	vector of all ones in \mathbb{R}^n
$\mathbf{e} \in \mathbb{R}^d$	vector of all ones in \mathbb{R}^d
$\mathbf{e}_j \in \mathbb{R}^n$	j -th standard unit basis vector in \mathbb{R}^n
$\mathbf{e}_i \in \mathbb{R}^n$	i -th standard unit basis vector in \mathbb{R}^d
$x^k \in \mathbb{R}^d$	the k -th iterate produced by Algorithm 1
$\mathbf{p} \in \mathbb{R}^d$	the vector (p_1, \dots, p_d)
$\mathbf{p}^t \in \mathbb{R}^d$	the vector (p_1^t, \dots, p_d^t)
$\mathbf{p} \in \mathbb{R}^n$	the vector (p_1, \dots, p_n)
$\mathbf{p}^t \in \mathbb{R}^n$	the vector (p_1^t, \dots, p_n^t)
$\mathbf{q} \in \mathbb{R}^n$	the vector (q_1, \dots, q_n)
$\mathbf{q}^t \in \mathbb{R}^n$	the vector (q_1^t, \dots, q_n^t)
$\mathbf{v} \in \mathbb{R}^n$	any vector for which (32) holds
$\langle x, y \rangle$	standard Euclidean inner product
$\ x\ $	standard Euclidean norm of vector x : $\ x\ = \langle x, x \rangle^{1/2}$
x^{-1}	elementwise inverse of x
g^k	estimator of the gradient $\nabla f(x^k)$ produced by Algorithm 1
Matrices in $\mathbb{R}^{d \times d}$, $\mathbb{R}^{d \times n}$ and $\mathbb{R}^{n \times n}$	
$\mathbf{I}_d \in \mathbb{R}^{d \times d}$	$d \times d$ identity matrix
$\mathbf{I}_n \in \mathbb{R}^{n \times n}$	$n \times n$ identity matrix
$\mathbf{G}(x) \in \mathbb{R}^{d \times n}$	the Jacobian matrix, i.e., $\mathbf{G}(x) = [\nabla f_1(x), \dots, \nabla f_n(x)]$
$\mathbf{J}^k \in \mathbb{R}^{d \times n}$	estimator of the Jacobian produced by Algorithm 1
$\mathbf{M}_j \in \mathbb{R}^{d \times d}$	smoothness matrix of f_j (if $\mathbf{M}_j = m^j \mathbf{I}_d$, then this specializes to m^j -smoothness)

$\mathbf{R} \in \mathbb{R}^{n \times n}$	a random matrix we use to multiply \mathbf{J} or \mathbf{G} from the right
$\mathbf{R}_R \in \mathbb{R}^{n \times n}$	the random matrix $\mathbf{R}_R := \sum_{j \in R} \mathbf{e}_j \mathbf{e}_j^\top$
$\mathbf{L} \in \mathbb{R}^{d \times d}$	a random matrix we use to multiply \mathbf{J} or \mathbf{G} from the left
$\mathbf{L}_L \in \mathbb{R}^{d \times d}$	the random matrix $\mathbf{L}_L := \sum_{i \in L} \mathbf{e}_i \mathbf{e}_i^\top$
$\langle \mathbf{X}, \mathbf{Y} \rangle$	trace inner product of matrices \mathbf{X} and \mathbf{Y} : $\langle \mathbf{X}, \mathbf{Y} \rangle := \text{Tr}(\mathbf{X}^\top \mathbf{Y})$
$\ \mathbf{X}\ $	Frobenius norm of matrix \mathbf{X} : $\ \mathbf{X}\ = \langle \mathbf{X}, \mathbf{X} \rangle^{1/2}$
$\mathbf{X} \circ \mathbf{Y}$	Hadamard product: $(\mathbf{X} \circ \mathbf{Y})_{ij} = \mathbf{X}_{ij} \mathbf{Y}_{ij}$
$\mathbf{X} \otimes \mathbf{Y}$	Kronecker product
$\mathbf{D}(x)$	diagonal matrix with vector x on the diagonal
$\mathbf{P} \in \mathbb{R}^{n \times n}$	Matrix such that $P_{jj'} = \mathbb{P}(j \in R, j' \in R)$
$\mathbf{P}^t \in \mathbb{R}^{n \times n}$	Matrix such that $P_{jj'} = \mathbb{P}(j \in R_t, j' \in R_t)$
$\mathbf{Q} \in \mathbb{R}^{d \times d}$	Matrix to define a norm in Lemma E.6
Linear operators $\mathbb{R}^{d \times n} \rightarrow \mathbb{R}^{d \times n}$	
\mathcal{A}	a generic linear operator
\mathcal{A}^*	the adjoint of \mathcal{A} : $\langle \mathcal{A}\mathbf{X}, \mathbf{Y} \rangle \equiv \langle \mathbf{X}, \mathcal{A}^*\mathbf{Y} \rangle$ for all $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d \times n}$
\mathcal{A}^\dagger	the Moore Penrose pseudoinverse of \mathcal{A}
$\text{Range}(\mathcal{A})$	image (range space) of \mathcal{A} : $\text{Range}(\mathcal{A}) := \{\mathcal{A}\mathbf{X} : \mathbf{X} \in \mathbb{R}^{d \times n}\}$
$\text{Range}(\mathcal{A})^\top$	orthogonal complement of $\text{Range}(\mathcal{A})$
$\text{Null}(\mathcal{A})$	kernel (null space) of \mathcal{A} : $\text{Null}(\mathcal{A}) := \{\mathbf{X} \in \mathbb{R}^{d \times n} : \mathcal{A}\mathbf{X} = 0\}$
\mathcal{I}	identity operator: $\mathcal{I}\mathbf{X} \equiv \mathbf{X}$
\mathcal{U}	any unbiased operator: $\mathbb{E}[\mathcal{U}\mathbf{X}] \equiv \mathbf{X}$, i.e., $\mathbb{E}[\mathcal{U}] \equiv \mathcal{I}$
\mathcal{S}	any random projection operator
\mathcal{M}	operator defined via $(\mathcal{M}\mathbf{X})_{:j} = \mathbf{M}_j \mathbf{X}_{:j}$
\mathcal{B}	(a technical) operator used to define the Lyapunov function (12)
\mathcal{R}	(a technical) operator such that $\mathbf{J}^k - \mathbf{G}(x^*) \in \text{Range}(\mathcal{R})$
Miscellaneous	
α	stepsize used in Algorithm 1
Γ	Random operator $\Gamma : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^d$ defined by $\Gamma\mathbf{X} = \mathcal{U}\mathbf{X}\mathbf{e}$
$\text{prox}_{\alpha\psi}(x)$	the proximal operator of ψ : $\text{prox}_{\alpha\psi}(x) := \arg\min_{u \in \mathbb{R}^d} \{\alpha\psi(u) + \frac{1}{2}\ u - x\ ^2\}$

D Additional Experiments

D.1 SEGA and SVRCD with importance sampling

In Sections H.2 and H.3 we develop an arbitrary (and thus importance in special case) sampling for SEGA, as well as new method SVRCD with arbitrary sampling. In this experiment, we compare them to its natural competitors – basic SEGA from [10] and proximal gradient descent. Since the main application of SEGA and SVRCD is in fact well studied SAGA/LSVRG (see Section R), we only consider artificial quadratic minimization with regularizer ψ being an indicator of the unit ball⁶:

$$f(x) = x^\top \mathbf{M}x - b^\top x, \quad \psi(x) = \begin{cases} x & 0 \leq 1 \\ \infty & \|x\| > 1 \end{cases}.$$

Specific choices of \mathbf{M}, b are given by Table 4. As both SEGA and SVRCD (from Section H.2 and H.3) require a diagonal smoothness matrix, we shall further consider vector m such that the upper bound $\mathbf{M} \preceq \mathbf{D}(m)$ holds. As the choice of m is not unique, we shall choose the one which minimizes $\sum_{i=1}^d m_i$ for importance sampling and $m = \lambda_{\max}(\mathbf{M})\mathbf{e}$ for uniform. Further, stepsize $\gamma = \frac{1}{4 \sum_{i=1}^d m_i}$ was chosen in each case. Figure 2 shows the results of this experiment. As theory suggests, importance sampling for both SEGA and SVRCD outperform both plain SEGA and proximal gradient always. The performance difference depends on the data; the closer \mathbf{M} is to a diagonal matrix with non-uniform elements, the larger stronger is the effect of importance sampling.

Type	\mathbf{M}	b
1	$\mathbf{D}(1.3^{[d]})$	γu
2	$\mathbf{D}((d, 1, 1, \dots, 1))$	γu
3	$\mathbf{D}(1.1^{[d]}) + \mathbf{N}\mathbf{N}^\top \frac{1.1^d}{1000d}, \mathbf{N} \sim N(0, \mathbf{I})$	γu
4	$\mathbf{N}\mathbf{N}^\top, \mathbf{N} \sim N(0, \mathbf{I})$	γu

Table 4: Four types of quadratic problems. We choose $u \sim N(0, \mathbf{I}_d)$, and γ to be such that $\|\gamma \mathbf{M}^{-1}u\| = 3/2$. Notation $c^{[d]}$ stands for a vector (c, c^2, \dots, c^d) .

D.2 SVRCD: Effect of ρ

In this experiment we demonstrate very broad range of ρ can be chosen to still attain almost best possible rate for SVRCD for problems from Table 4 and m, γ as described in Section D.1 Results can be found in Figure 3. They indeed show that in many cases, varying ρ from $\frac{1}{n}$ down to $\frac{2\lambda_{\min}(\mathbf{M})}{\sum_{i=1}^d m_i}$ does not influence the complexity significantly. However, too small ρ leads to significantly slower convergence. Note that those findings are in accord with Corollary H.4. Similar results were shown in [15] for LSVRG.

D.3 ISAEGA

In this section we test a simple version of ISAEGA (Algorithm 16)⁷. As mentioned, ISAEGA is an algorithm for distributed optimization which, at each iteration, computes a subset of partial derivatives of stochastic gradient on each machine, and constructs corresponding Jacobian estimate and stochastic gradient.

For simplicity, we consider only the simple version which assumes $\mathbf{M}_j = m\mathbf{I}_d$ for all j (i.e. we do not do importance sampling), and we suppose that $|R_t| = 1$ always for all t (i.e. each machine always

⁶In such case, proximal operator of ψ becomes a projection onto the unit ball.

⁷The full description of ISAEGA, together with convergence guarantees are provided in Section M.3

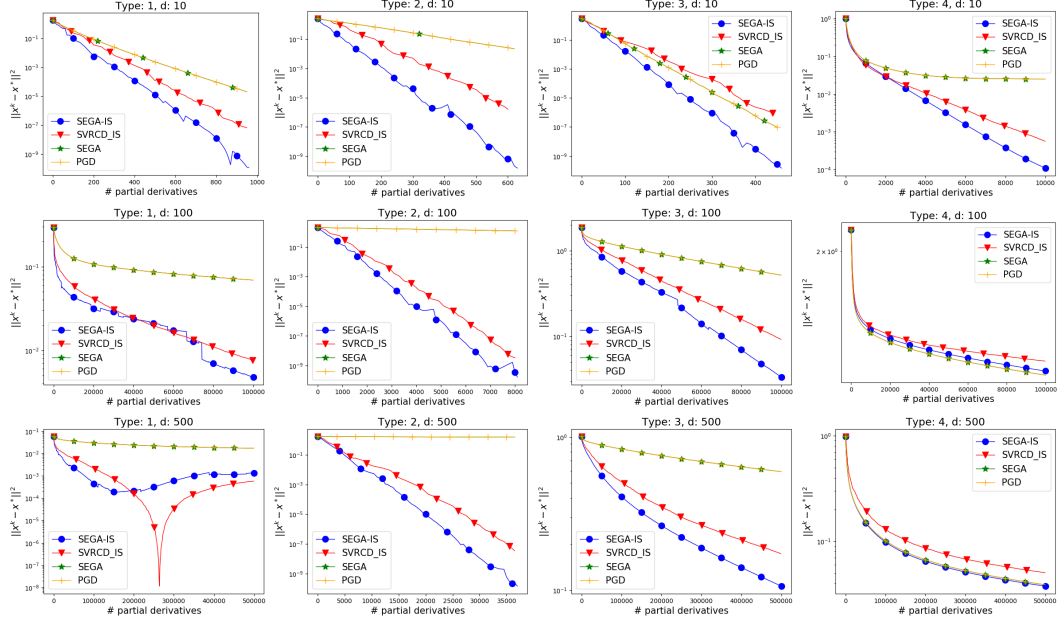


Figure 2: Comparison of SEGA-AS, SVRCD-AS, SEGA and proximal gradient on 4 quadratic problems given by Table 4. SEGA-AS, SVRCD-AS and SEGA compute single partial derivative each iteration (SVRCD computes all of them with probability ρ), SEGA-AS, SVRCD-AS with probabilities proportional to diagonal of \mathbf{M} .

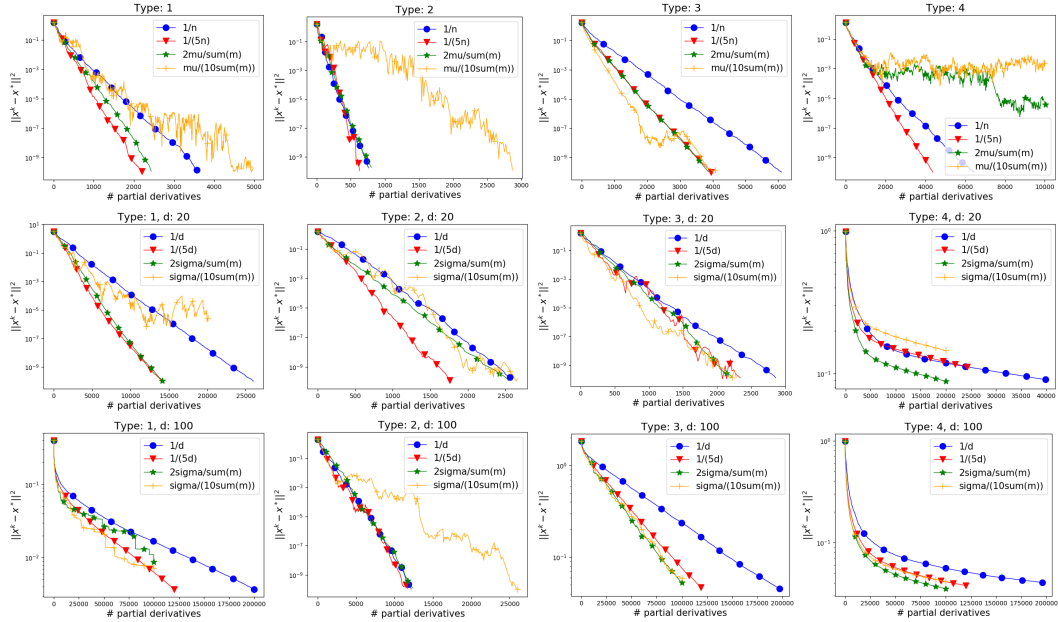


Figure 3: The effect of ρ on convergence rate of SVRCD on quadratic problems from Table 4. In every case, probabilities were chosen proportionally to the diagonal of \mathbf{M} and only a single partial derivative is evaluated in \mathcal{S} .

looks at a single function from the local finite sum). Further, we consider $\psi(x) = 0$. Corollary M.3 shows that, if the condition number of the problem is not too small, ISAEGA with $|L_t| \approx \frac{1}{T}$ (where T is a number of parallel units) enjoys, up to small constant factor, same rate as SAGA (which is, under a

convenient smoothness, the same rate as the convergence rate of gradient descent). Thus, ISAEGA scales linearly in terms of partial derivative complexity in parallel setup. In other words, given that we have twice more workers, each of them can afford to evaluate twice less partial derivatives⁸. The experiments we propose aim to verify this claim.

We consider 12 regularized logistic regression (for the binary classification). In particular,

$$\forall j : f_j(x) := \log(1 + \exp(\mathbf{A}_{j,:}x \cdot y_i)) + \frac{\lambda}{2}\|x\|^2,$$

where $\mathbf{A} \in \mathbb{R}^{n \times d}$ is a data matrix, $y \in \{-1, 1\}^n$ is a vector of labels and $\lambda \geq 0$ is the regularization parameter. Both \mathbf{A}, y are provided from LibSVM [2] datasets: a1a, a9a, w1a, w8a, gisette, madelon, phishing and mushrooms. Further, \mathbf{A} was normalized such that $\|\mathbf{A}_{j,:}\|^2 = 1$. Next, it is known that f_j is $(\frac{1}{4} + \lambda)$ -smooth, convex, while f is λ -strongly convex. Therefore, as a stepsize for all versions of ISAEGA, we set $\gamma = \frac{1}{6\lambda + \frac{3}{2}}$ (this is an approximation of theoretical stepsize).

In each experiment, we compare 4 different setups for ISEAGA – given by 4 different values of T . Given a value of T , we set $|L_t| = \frac{1}{T}$ for all t . Further, we always sample L_t uniformly. The results are presented in Figure 4. Indeed, we observe the almost perfect parallel linear scaling.

For completeness, we provide dataset sized in Table 5.

Name	n	d
a1a	1605	123
a9a	32561	123
w1a	2477	300
w8a	49749	300
gisette	6000	5000
madelon	2000	500
phishing	11055	68
mushrooms	8124	112

Table 5: Table of LibSVM data used for our experiments.

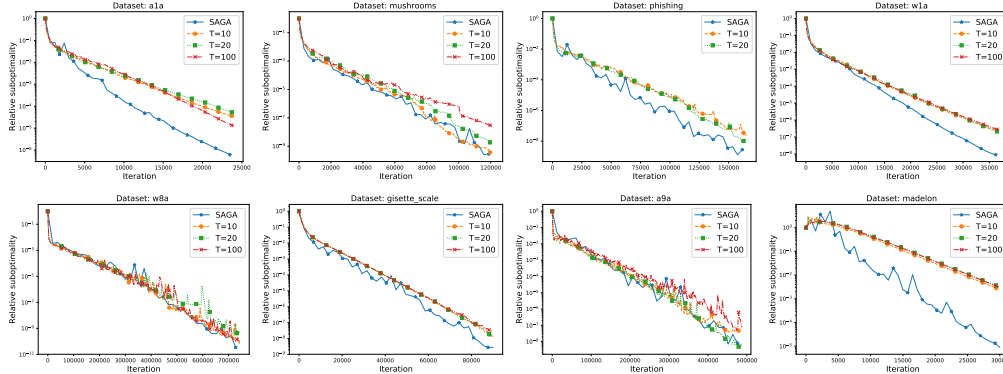


Figure 4: ISAEGA applied on LIBSVM [2] datasets with $\lambda = 4 \cdot 10^{-5}$. Axis y stands for relative suboptimality, i.e. $\frac{f(x^k) - f(x^*)}{f(x^k) - f(x^0)}$.

⁸Practical implications of the method are further explained in [21].

D.4 LSVRG with importance sampling

As mentioned, one of the contributions of this work is LSVRG with arbitrary sampling. In this section, we demonstrate that designing a good sampling can yield a significant speedup in practice. We consider logistic regression problem on LibSVM [2] data, as described in Section D.3. However, since LibSVM data are normalized, we pre-multiply each row of the data matrix by a random scaling factor. In particular, the scaling factors are proportional to l^2 where l is sampled uniformly from $[1000]$ such that the Frobenius norm of the data matrix is n . For the sake of simplicity, consider case $\lambda = 0$.

Choice vector v . Note that since $\mathbf{M}_j = \mathbf{A}_{j:}^\top \mathbf{A}_{j:}$, the following claim must hold: *Consider fixed v . Then if (32) holds for any set of vector $\{h_j\}_{j=1}^n$ such that h_j is parallel to $\mathbf{A}_{j:}$, then (32) holds for any set of vector $\{h_j\}_{j=1}^n$.* Thus, we can set $h_j = c_j \mathbf{A}_{j:}^\top / \|\mathbf{A}_{j:}\|$ without loss of generality. Thus, $\mathbf{M}_j^{\frac{1}{2}} h_j = c_j \mathbf{A}_{j:}^\top$, and (32) becomes equivalent to $\mathbf{P} \circ (\mathbf{A}^\top \mathbf{A}) \preceq \mathbf{D}(p \circ v)$ where $\mathbf{P}_{jj'} = \mathbb{P}(j \in R, j' \in R)$. Note that this is exactly *expected separable overapproximation (ESO)* for coordinate descent [29]. Thus we choose vector v to be proportional to p such that $\mathbf{P} \circ (\mathbf{A}^\top \mathbf{A}) \preceq \mathbf{D}(p \circ v)$ holds (as proposed in [11]). In order to compute the scaling constant, one needs to evaluate maximum eigenvalue of PSD $n \times n$ matrix, which is of $\mathcal{O}(n^2)$ cost. We do so in the experiments. Note that there is a suboptimal, but cheaper way to obtain v described in [27]. Lastly, if $\lambda > 0$, we set v such that $\mathbf{P} \circ (\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I}) \preceq \mathbf{D}(p \circ v)$.

Choice of probabilities. In order to be fair, we only compare methods where $\mathbb{E}[|R|] = \tau$. For the case $\tau = 1$, we consider a sampling such that $|R| = 1$ according to a given probability vector p . For uniform sampling, we have $p = n^{-1} \mathbf{e}$, while for importance sampling, we set $p_j = \frac{\lambda_{\max}(\mathbf{M}_j)}{\sum_{j'=1}^n \lambda_{\max}(\mathbf{M}_{j'})}$. In the case $\tau > 1$, we consider independent sampling from [11]. In particular, $\mathbb{P}(j \in R) = p_j$ with $\sum p_j = \tau$ and binary random variables ($j \in R$) are jointly independent. For uniform sampling we have $p = \tau n^{-1} \mathbf{e}$. For importance sampling, probability vector p is chosen such that $p_j = \frac{\lambda_{\max}(\mathbf{M}_j)}{\varrho + \lambda_{\max}(\mathbf{M}_j)}$, where ϱ is such that $\sum p_j = \tau$. The mentioned sampling was proven to be superior over uniform minibatching in [11]. Next, stepsize $\gamma = \frac{1}{6} \min_j \frac{np_j}{v_j}$ was chosen for all methods.

Lastly, $\rho = \frac{1}{2n}$ was chosen for LSVRG. The results are presented in Figures 5 and 6 (a subset of the results was already presented in Figure 1).

In all cases, LSVRG with importance sampling was the fastest method. As provided theory suggests, it outperformed methods with importance sampling especially significantly for small τ ; and the larger τ , the smaller the effect of importance sampling is. However, our experiments indicate the superiority of LSVRG to SAGA in the importance sampling setup. In particular, stepsize $\gamma = \frac{1}{6} \min_j \frac{np_j}{v_j}$ is often too large for SAGA. Note that both optimal stepsize and optimal probabilities require the prior knowledge of the quasi strong convexity constant σ^9 which is, in our case unknown (see the importance serial sampling proposed in [8], and SAGA is more sensitive to that choice. One can still estimate it as λ , however, this would yield suboptimal performance as well.

⁹Or more generally, strong growth constant, see Section S

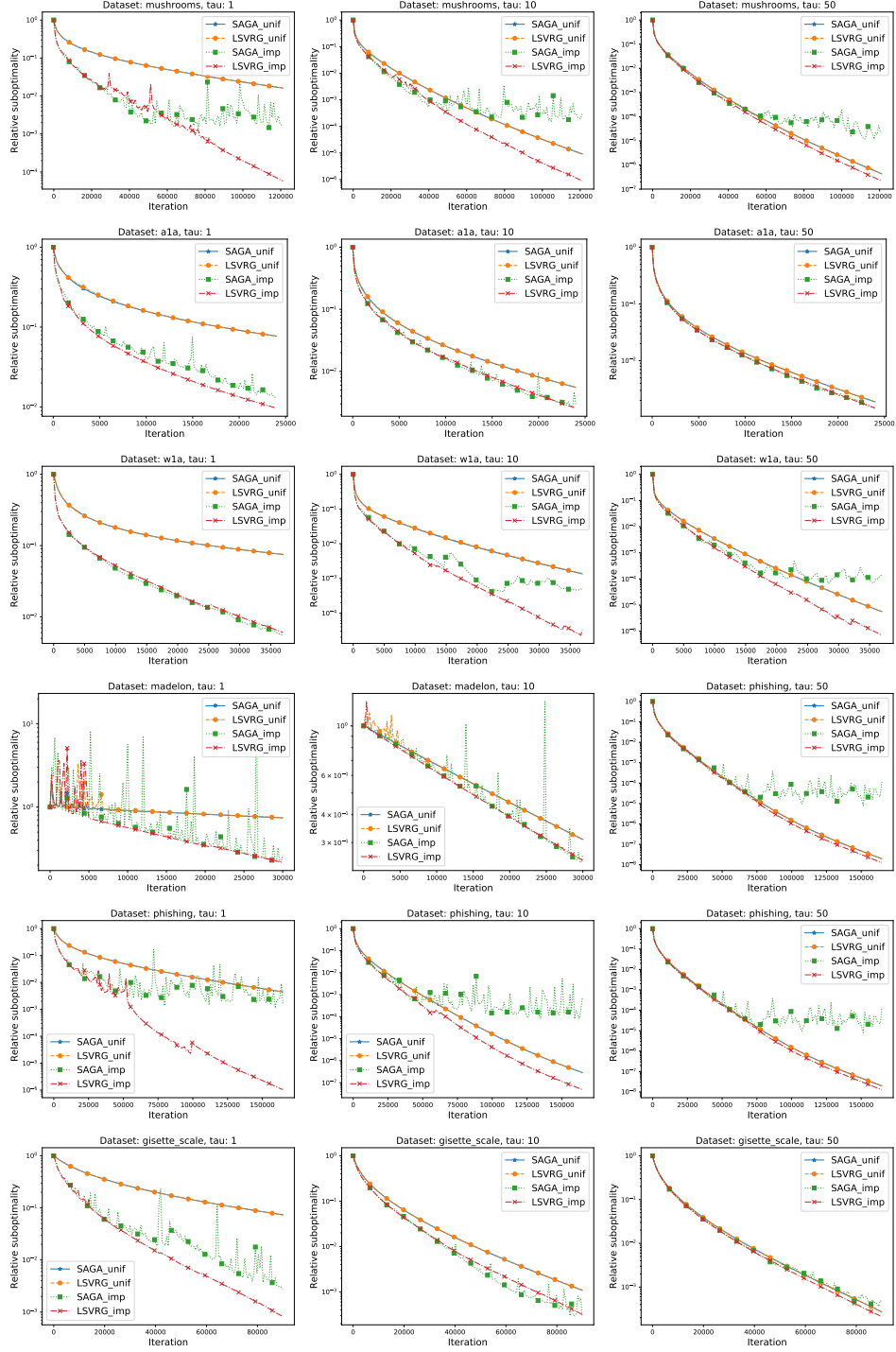


Figure 5: LSVRG applied on LIBSVM [2] datasets with $\lambda = 10^{-5}$. Axis y stands for relative suboptimality, i.e. $\frac{f(x^k) - f(x^*)}{f(x^k) - f(x^0)}$.

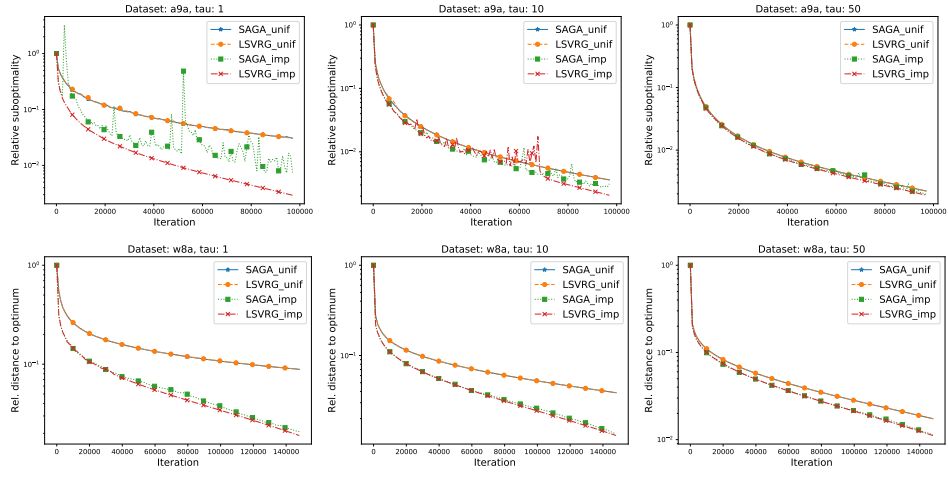


Figure 6: LSVRG applied on LIBSVM [2] datasets. For a9a, $\lambda = 0$ and $\rho = \frac{1}{n}$ was chosen; for w8a, $\lambda = 10^{-8}$ and $\rho = \frac{3}{n}$ was chosen. Axis y stands for relative suboptimality, i.e. $\frac{f(x^k) - f(x^*)}{f(x^k) - f(x^0)}$.

E Several Lemmas

E.1 Existence lemma

Lemma E.1. Suppose that $\mathcal{X} \in \text{Range}(\mathcal{M})$. Denote $\Gamma(\mathbf{X}) := \mathcal{U}\mathbf{X}\mathbf{e}$. Suppose that $\mathbb{E} \left[\left(\Gamma \mathcal{M}^{\frac{1}{2}} \right)^* \Gamma \mathcal{M}^{\frac{1}{2}} \right]$ exists and $\lambda_{\min}(\mathbb{E}[\mathcal{S}]) > 0$. Then, there are $\alpha > 0$ and \mathcal{B} such that (13) and (14) hold. Moreover, inequalities (13), (14) hold for $\alpha = 0, \mathcal{B} = 0$ without any extra assumptions.

Proof: Consider only α, \mathcal{B} such that $\alpha < \lambda_{\min}(\mathbb{E}[\mathcal{S}]) \sigma^{-1}$, $\lambda_{\min}(\mathcal{B}^* \mathcal{B}) > 0$, $\lambda_{\max}(\mathcal{B}^* \mathcal{B}) < \infty$. Let $\mathbf{Y} = \mathcal{M}^{\dagger \frac{1}{2}} \mathbf{X}$. Thus we have $\mathbb{E} \left[\|\mathcal{U}\mathbf{X}\mathbf{e}\|^2 \right] \leq \frac{2\alpha}{n^2} \|\mathbf{Y}\|^2 \lambda_{\max} \mathbb{E} \left[\left(\Gamma \mathcal{M}^{\frac{1}{2}} \right)^* \Gamma \mathcal{M}^{\frac{1}{2}} \right]$.

Thus

$$\begin{aligned} (1 - \alpha\sigma) \|\mathcal{B}\mathbf{Y}\|^2 - \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}\mathbf{Y} \right\|^2 &= \langle (\mathcal{B}\mathbf{Y})^\top, (\mathbb{E}[\mathcal{S}] - \alpha\sigma\mathcal{I}) \mathcal{B}\mathbf{Y} \rangle \\ &\geq (\lambda_{\min}(\mathbb{E}[\mathcal{S}]) - \alpha\sigma\mathcal{I}) \|\mathcal{B}\mathbf{Y}\|^2 \\ &\geq (\lambda_{\min}(\mathbb{E}[\mathcal{S}]) - \alpha\sigma\mathcal{I}) \lambda_{\min}(\mathcal{B}^* \mathcal{B}) \|\mathbf{Y}\|^2. \end{aligned}$$

Therefore, to have (13), it suffices to set

$$\alpha \leq \frac{\lambda_{\min}(\mathbb{E}[\mathcal{S}]) \lambda_{\min}(\mathcal{B}^* \mathcal{B})}{\alpha\sigma \lambda_{\min}(\mathcal{B}^* \mathcal{B}) + \frac{2}{n^2} \lambda_{\max} \left[\mathbb{E} \left[\left(\Gamma \mathcal{M}^{\frac{1}{2}} \right)^* \Gamma \mathcal{M}^{\frac{1}{2}} \right] \right]}.$$

Similarly, to satisfy (14), it suffices to have

$$\frac{2\alpha}{n} \lambda_{\max} \left(\mathbb{E} \left[\left(\Gamma \mathcal{M}^{\frac{1}{2}} \right)^* \Gamma \mathcal{M}^{\frac{1}{2}} \right] \right) + n \lambda_{\min}(\mathbb{E}[\mathcal{S}]) \lambda_{\max}(\mathcal{B}^* \mathcal{B}) \leq 1.$$

A valid choice is for example \mathcal{B} such that

$$\lambda_{\max}(\mathcal{B}^* \mathcal{B}) \leq \frac{1}{2n \lambda_{\min}(\mathbb{E}[\mathcal{S}])}, \quad \alpha \leq \frac{1}{\frac{1}{n} \lambda_{\max} \left(\mathbb{E} \left[\Gamma \left(\mathcal{M}^{\frac{1}{2}} \right)^* \Gamma \mathcal{M}^{\frac{1}{2}} \right] \right)}.$$

E.2 Smoothness lemmas

Let $h : \mathbb{R}^d \rightarrow \mathbb{R}$ be a differentiable and convex function. The Bregman distance of x and y with respect to h is defined by

$$D_h(x, y) := h(x) - h(y) - \langle \nabla h(y), x - y \rangle. \quad (15)$$

Lemma E.2 (Lemma A.1 from [10]). Suppose that function $h : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex and \mathbf{M} -smooth, where $\mathbf{M} \succeq 0$. Then

$$D_h(x, y) \geq \frac{1}{2} \|\nabla h(y) - \nabla h(x)\|_{\mathbf{M}^\dagger}^2, \quad \forall x, y \in \mathbb{R}^d. \quad (16)$$

Further,

$$\langle \nabla h(x) - \nabla h(y), x - y \rangle \geq \|\nabla h(x) - \nabla h(y)\|_{\mathbf{M}^\dagger}^2. \quad (17)$$

Proof: Fix y and consider the function $\phi(x) := h(x) - \langle \nabla h(y), x \rangle$. Clearly, ϕ is \mathbf{M} -smooth, and hence

$$\phi(x + d) \leq \phi(x) + \langle \nabla \phi(x), d \rangle + \frac{1}{2} \|d\|_{\mathbf{M}}^2, \quad \forall x, d \in \mathbb{R}^d. \quad (18)$$

Moreover, since h is convex, ϕ is convex, non-negative and is minimized at y . Letting $t = \nabla h(x) - \nabla h(y)$, this implies that

$$\begin{aligned}
\phi(y) &\leq \phi(x - \mathbf{M}^\dagger t) \\
&\stackrel{(18)}{\leq} \phi(x) - \langle \nabla \phi(x), \mathbf{M}^\dagger t \rangle + \frac{1}{2} \|\mathbf{M}^\dagger t\|_{\mathbf{M}}^2 \\
&= \phi(x) - \langle t, \mathbf{M}^\dagger t \rangle + \frac{1}{2} \|\mathbf{M}^\dagger t\|_{\mathbf{M}}^2 \\
&= \phi(x) - \frac{1}{2} \|t\|_{\mathbf{M}^\dagger}^2,
\end{aligned}$$

which is equivalent to (16). In the last step we have used the identities $(\mathbf{M}^\dagger)^\top = (\mathbf{M}^\top)^\dagger = \mathbf{M}^\dagger$ and $\mathbf{M}^\dagger \mathbf{M} \mathbf{M}^\dagger = \mathbf{M}^\dagger$.

To show (17), it suffices to sum inequality 16 applied on vector pairs (x, y) and (y, x) . □

Lemma E.3. *Let (11) hold. That is, assume that function f_j are convex and \mathbf{M}_j -smooth. Then*

$$D_{f_i}(x, y) \geq \frac{1}{2} \|\nabla f_j(x) - \nabla f_j(y)\|_{\mathbf{M}_j^\dagger}^2, \quad \forall x, y \in \mathbb{R}^d. \quad (19)$$

If $x - y \in \text{Null}(\mathbf{M}_j)$, then

$$(i) \quad f_j(x) = f_j(y) + \langle \nabla f_j(y), x - y \rangle, \quad (20)$$

$$(ii) \quad \nabla f_j(x) - \nabla f_j(y) \in \text{Null}(\mathbf{M}_j), \quad (21)$$

$$(iii) \quad \langle \nabla f_j(x) - \nabla f_j(y), x - y \rangle = 0. \quad (22)$$

If, in addition, f_j is bounded below, then $\nabla f_j(x) \in \text{Range}(\mathbf{M}_j)$ for all x .

Proof: Inequality (19) follows by applying Lemma E.3 for $h = f_j$ and $\mathbf{M} = \mathbf{M}_j$. Identity (20) is a direct consequence of (11). Combining (19) and (20), we get $0 \geq \frac{1}{2} \|\nabla f_j(x) - \nabla f_j(y)\|_{\mathbf{M}_j^\dagger}^2$, which implies that

$$\nabla f_j(x) - \nabla f_j(y) \in \text{Null}(\mathbf{M}_j^\dagger) = \text{Null}(\mathbf{M}_j^\top) = \text{Null}(\mathbf{M}_j), \quad (23)$$

recovering (21). By adding two copies of (20) (with the roles of x and y exchanged), we get (22). Finally, if f_j is bounded below, then in view of (20) there exists $c \in \mathbb{R}$ such that,

$$c \leq \inf_{x \in y + \text{Null}(\mathbf{M}_j)} f_j(x) \stackrel{(20)}{=} \inf_{x \in y + \text{Null}(\mathbf{M}_j)} f_j(y) + \langle \nabla f_j(y), x - y \rangle.$$

This implies that $\nabla f_j(y) \in \text{Range}(\mathbf{M}_j^\top) = \text{Range}(\mathbf{M}_j)$. □

Lemma E.4. *Assume f is twice continuously differentiable. Then $\mathbf{G}(x) - \mathbf{G}(y) \in \text{Range}(\mathcal{M})$ for all $x, y \in \mathbb{R}^d$.*

Proof: For $\mathbf{G}(x) - \mathbf{G}(y) \in \text{Range}(\mathcal{M})$, it suffices to show that $\nabla f_j(x) - \nabla f_j(y) \in \text{Range}(\mathbf{M}_j)$. Without loss of generality, suppose that $f(z, w)$ (for $x = [z, w]$) is such that $f(z, \cdot)$ is linear (for fixed z ; from (20)) and $f(\cdot, w)$ is \mathbf{M}' smooth for full rank \mathbf{M}' . Note that

$$0 \preceq \nabla^2 f(x) = \begin{pmatrix} \nabla_{ww}^2 f(w, z) & \nabla_{wz}^2 f(w, z) \\ \nabla_{zw}^2 f(w, z) & \nabla_{zz}^2 f(w, z) \end{pmatrix} = \begin{pmatrix} \nabla_{ww}^2 f(w, z) & \nabla_{wz}^2 f(w, z) \\ \nabla_{zw}^2 f(w, z) & 0 \end{pmatrix}.$$

Since every submatrix of the above must be positive definite, it is easy to see that we must have both $\nabla_{wz}^2 f(w, z) = 0$, $\nabla_{zw}^2 f(w, z) = 0$. This, however, means that $f(w, z)$ is separable in z, w . Therefore indeed $\nabla f_j(x) - \nabla f_j(y) \in \text{Range}(\mathbf{M}_j)$ for all $x, y \in \mathbb{R}^d$ and all $j \in [n]$.

E.3 Projection lemma

In the next lemma, we establish some basic properties of the interaction of the random projection matrices \mathcal{S} and $\mathcal{I} - \mathcal{S}$ with various matrices, operators, and norms.

Lemma E.5. *Let \mathcal{S} be a random projection operator and \mathcal{A} any deterministic linear operator commuting with \mathcal{S} , i.e., $\mathcal{A}\mathcal{S} = \mathcal{S}\mathcal{A}$. Further, let $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d \times n}$ and define $\mathbf{Z} = (\mathcal{I} - \mathcal{S})\mathbf{X} + \mathcal{S}\mathbf{Y}$. Then*

- (i) $\mathcal{A}\mathbf{Z} = (\mathcal{I} - \mathcal{S})\mathcal{A}\mathbf{X} + \mathcal{S}\mathcal{A}\mathbf{Y}$,
- (ii) $\|\mathcal{A}\mathbf{Z}\|^2 = \|(\mathcal{I} - \mathcal{S})\mathcal{A}\mathbf{X}\|^2 + \|\mathcal{S}\mathcal{A}\mathbf{Y}\|^2$,
- (iii) $\mathbb{E} \left[\|\mathcal{A}\mathbf{Z}\|^2 \right] = \|(\mathcal{I} - \mathbb{E}[\mathcal{S}])^{1/2} \mathcal{A}\mathbf{X}\|^2 + \left\| \mathbb{E}[\mathcal{S}]^{1/2} \mathcal{A}\mathbf{Y} \right\|^2$, where the expectation is with respect to \mathcal{S} .

Proof: Part (i) follows by noting that \mathcal{A} commutes with $\mathcal{I} - \mathcal{S}$. Part (ii) follows from (i) by expanding the square, and noticing that $(\mathcal{I} - \mathcal{S})\mathcal{S} = 0$. Part (iii) follows from (ii) after using the definition of the Frobenius norm, i.e., $\|\mathbf{M}\|^2 = \text{Tr}(\mathbf{M}^\top \mathbf{M})$, the identities $(\mathcal{I} - \mathcal{S})^2 = \mathcal{I} - \mathcal{S}$, $\mathcal{S}^2 = \mathcal{S}$, and taking expectation on both sides. \square

E.4 Decomposition lemma

In the next lemma, we give a bound on the expected squared distance of the gradient estimator g^k from $\nabla f(x^*)$.

Lemma E.6. *For any $\mathbf{Q} \in \mathbb{R}^{d \times d}$, $\mathbf{Q} \succeq 0$ and all $k \geq 0$ we have*

$$\mathbb{E} \left[\|g^k - \nabla f(x^*)\|_{\mathbf{Q}}^2 \right] \leq \frac{2}{n^2} \mathbb{E} \left[\|\mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*)) \mathbf{e}\|_{\mathbf{Q}}^2 \right] + \frac{2}{n^2} \mathbb{E} \left[\|\mathcal{U}(\mathbf{J}^k - \mathbf{G}(x^*)) \mathbf{e}\|_{\mathbf{Q}}^2 \right]. \quad (24)$$

Proof: In view of (7) and since $\nabla f(x^*) = \frac{1}{n} \mathbf{G}(x^*) \mathbf{e}$, we have

$$g^k - \nabla f(x^*) = \underbrace{\frac{1}{n} \mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*)) \mathbf{e}}_a + \underbrace{\frac{1}{n} (\mathbf{J}^k - \mathbf{G}(x^*)) \mathbf{e} - \frac{1}{n} \mathcal{U}(\mathbf{J}^k - \mathbf{G}(x^*)) \mathbf{e}}_b. \quad (25)$$

Applying the bound $\|a + b\|_{\mathbf{Q}}^2 \leq 2\|a\|_{\mathbf{Q}}^2 + 2\|b\|_{\mathbf{Q}}^2$ to (25) and taking expectations, we get

$$\begin{aligned} \mathbb{E} \left[\|g^k - \nabla f(x^*)\|_{\mathbf{Q}}^2 \right] &\leq \mathbb{E} \left[\frac{2}{n^2} \|\mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*)) \mathbf{e}\|_{\mathbf{Q}}^2 \right] \\ &\quad + \mathbb{E} \left[\frac{2}{n^2} \|(\mathbf{J}^k - \mathbf{G}(x^*)) \mathbf{e} - \mathcal{U}(\mathbf{J}^k - \mathbf{G}(x^*)) \mathbf{e}\|_{\mathbf{Q}}^2 \right] \\ &= \frac{2}{n^2} \mathbb{E} \left[\|\mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*)) \mathbf{e}\|_{\mathbf{Q}}^2 \right] \\ &\quad + \frac{2}{n^2} \mathbb{E} \left[\|(\mathcal{I} - \mathcal{U})(\mathbf{J}^k - \mathbf{G}(x^*)) \mathbf{e}\|_{\mathbf{Q}}^2 \right]. \end{aligned}$$

It remains to note that

$$\begin{aligned} \mathbb{E} \left[\|(\mathcal{I} - \mathcal{U})(\mathbf{J}^k - \mathbf{G}(x^*)) \mathbf{e}\|_{\mathbf{Q}}^2 \right] &= \mathbb{E} \left[\|\mathcal{U}(\mathbf{J}^k - \mathbf{G}(x^*)) \mathbf{e}\|_{\mathbf{Q}}^2 \right] - \|(\mathbf{J}^k - \mathbf{G}(x^*)) \mathbf{e}\|_{\mathbf{Q}}^2 \\ &\leq \mathbb{E} \left[\|\mathcal{U}(\mathbf{J}^k - \mathbf{G}(x^*)) \mathbf{e}\|_{\mathbf{Q}}^2 \right]. \end{aligned}$$

\square

F Proof of Theorem 5.2

For simplicity of notation, in this proof, all expectations are conditional on x^k , i.e., the expectation is taken with respect to the randomness of g^k .

Since

$$x^* = \text{prox}_{\alpha\psi}(x^* - \alpha\nabla f(x^*)), \quad (26)$$

and since the prox operator is non-expansive, we have

$$\begin{aligned} \mathbb{E} \left[\|x^{k+1} - x^*\|^2 \right] &\stackrel{(26)}{=} \mathbb{E} \left[\|\text{prox}_{\alpha\psi}(x^k - \alpha g^k) - \text{prox}_{\alpha\psi}(x^* - \alpha\nabla f(x^*))\|^2 \right] \\ &\leq \mathbb{E} \left[\|x^k - x^* - \alpha(g^k - \nabla f(x^*))\|^2 \right] \\ &\stackrel{(8)}{=} \|x^k - x^*\|^2 - 2\alpha \langle \nabla f(x^k) - \nabla f(x^*), x^k - x^* \rangle \\ &\quad + \alpha^2 \mathbb{E} \left[\|g^k - \nabla f(x^*)\|^2 \right] \\ &\stackrel{(10)+(15)}{\leq} (1 - \alpha\sigma) \|x^k - x^*\|^2 + \alpha^2 \mathbb{E} \left[\|g^k - \nabla f(x^*)\|^2 \right] \\ &\quad - 2\alpha D_f(x^k, x^*). \end{aligned} \quad (27)$$

Since $f(x) = \frac{1}{n} \sum_{j=1}^n f_j(x)$, in view of (15) and (19) we have

$$\begin{aligned} D_f(x^k, x^*) &\stackrel{(15)}{=} \frac{1}{n} \sum_{j=1}^n D_{f_j}(x^k, x^*) \stackrel{(19)}{\geq} \frac{1}{2n} \sum_{j=1}^n \|\nabla f_j(x^k) - \nabla f_j(x^*)\|_{\mathbf{M}_j^\dagger}^2 \\ &= \frac{1}{2n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2. \end{aligned} \quad (28)$$

By combining (27) and (28), we get

$$\begin{aligned} \mathbb{E} \left[\|x^{k+1} - x^*\|^2 \right] &\leq (1 - \alpha\sigma) \|x^k - x^*\|^2 + \alpha^2 \mathbb{E} \left[\|g^k - \nabla f(x^*)\|^2 \right] \\ &\quad - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2. \end{aligned}$$

Next, applying Lemma E.6 with $\mathbf{Q} = \mathbf{I}$ leads to the estimate

$$\begin{aligned} \mathbb{E} \left[\|x^{k+1} - x^*\|^2 \right] &\leq (1 - \alpha\sigma) \|x^k - x^*\|^2 - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2 \\ &\quad + \frac{2\alpha^2}{n^2} \mathbb{E} \left[\|\mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*)) \mathbf{e}\|^2 \right] \\ &\quad + \frac{2\alpha^2}{n^2} \mathbb{E} \left[\|\mathcal{U}(\mathbf{J}^k - \mathbf{G}(x^*)) \mathbf{e}\|^2 \right]. \end{aligned} \quad (29)$$

In view of (9), we have $\mathbf{J}^{k+1} = (\mathcal{I} - \mathcal{S})\mathbf{J}^k + \mathcal{S}\mathbf{G}(x^k)$, whence

$$\underbrace{\mathbf{J}^{k+1} - \mathbf{G}(x^*)}_{\mathbf{Z}} = (\mathcal{I} - \mathcal{S}) \underbrace{(\mathbf{J}^k - \mathbf{G}(x^*))}_{\mathbf{X}} + \mathcal{S} \underbrace{(\mathbf{G}(x^k) - \mathbf{G}(x^*))}_{\mathbf{Y}}. \quad (30)$$

Since, by assumption, both \mathcal{B} and $\mathcal{M}^{\dagger \frac{1}{2}}$ commute with \mathcal{S} , so does their composition $\mathcal{A} := \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}}$. Applying Lemma E.5, we get

$$\begin{aligned} \mathbb{E} \left[\left\| \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^{k+1} - \mathbf{G}(x^*)) \right\|^2 \right] &= \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^k - \mathbf{G}(x^*)) \right\|^2 \\ &\quad + \left\| \mathbb{E}[\mathcal{S}]^{\frac{1}{2}} \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2. \end{aligned} \quad (31)$$

Adding α -multiple of (31) to (29) yields

$$\begin{aligned}
& \mathbb{E} \left[\|x^{k+1} - x^*\|^2 \right] + \alpha \mathbb{E} \left[\left\| \mathcal{B} \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^{k+1} - \mathbf{G}(x^*)) \right\|^2 \right] \\
& \leq (1 - \alpha\sigma) \|x^k - x^*\|^2 + \frac{2\alpha^2}{n^2} \mathbb{E} \left[\left\| \mathcal{U} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \mathbf{e} \right\|^2 \right] \\
& \quad + \frac{2\alpha^2}{n^2} \mathbb{E} \left[\left\| \mathcal{U} (\mathbf{J}^k - \mathbf{G}(x^*)) \mathbf{e} \right\|^2 \right] + \alpha \left\| (\mathcal{I} - \mathbb{E} [\mathcal{S}])^{\frac{1}{2}} \mathcal{B} \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^k - \mathbf{G}(x^*)) \right\|^2 \\
& \quad + \alpha \left\| \mathbb{E} [\mathcal{S}]^{\frac{1}{2}} \mathcal{B} \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2 - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2 \\
& \stackrel{(13)}{\leq} (1 - \alpha\sigma) \|x^k - x^*\|^2 + (1 - \alpha\sigma)\alpha \left\| \mathcal{B} \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^k - \mathbf{G}(x^*)) \right\|^2 \\
& \quad + \frac{2\alpha^2}{n^2} \mathbb{E} \left[\left\| \mathcal{U} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \mathbf{e} \right\|^2 \right] + \alpha \left\| \mathbb{E} [\mathcal{S}]^{\frac{1}{2}} \mathcal{B} \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2 \\
& \quad - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2 \\
& \stackrel{(14)}{\leq} (1 - \alpha\sigma) \left(\|x^k - x^*\|^2 + \alpha \left\| \mathcal{B} \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^k - \mathbf{G}(x^*)) \right\|^2 \right).
\end{aligned}$$

Above, we have used (13) with $\mathbf{X} = \mathbf{J}^k - \mathbf{G}(x^*)$ and (14) with $\mathbf{X} = \mathbf{G}(x^k) - \mathbf{G}(x^*)$. \square

G Special Cases: SAGA-like Methods

G.1 Basic variant of SAGA [3]

Suppose that for all j , f_j is m -smooth (i.e., $\mathbf{M}_j = m\mathbf{I}_d$). To recover basic SAGA [3], consider the following choice of random operators \mathcal{S}, \mathcal{U} :

$$(\forall j) \text{ with probability } \frac{1}{n} : \quad \mathcal{S}\mathbf{X} = \mathbf{X}e_j e_j^\top \quad \text{and} \quad \mathcal{U}\mathbf{X} = \mathbf{X}n e_j e_j^\top.$$

The resulting algorithm is stated as Algorithm 2. Further, as a direct consequence of Theorem 5.2, convergence rate of SAGA (Algorithm 2) is presented in Corollary G.1.

Algorithm 2 SAGA [3]

Require: learning rate $\alpha > 0$, starting point $x^0 \in \mathbb{R}^d$

Set $\psi_j^0 = x^0$ for each $j \in \{1, 2, \dots, n\}$

for $k = 0, 1, 2, \dots$ **do**

 Sample $j \in [n]$ uniformly at random

 Set $\phi_j^{k+1} = x^k$ and $\phi_i^{k+1} = \phi_i^k$ for $i \neq j$

$$g^k = \nabla f_j(\phi_j^{k+1}) - \nabla f_j(\phi_j^k) + \frac{1}{n} \sum_{i=1}^n \nabla f_i(\phi_i^k)$$

$$x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$$

end for

Corollary G.1 (Convergence rate of SAGA). *Let $\alpha = \frac{1}{4m + \sigma n}$. Then, iteration complexity of Algorithm 2 (proximal SAGA) is $(4\frac{m}{\sigma} + n) \log \frac{1}{\epsilon}$.*

G.2 SAGA with arbitrary sampling

In contrast to Section G.1, here we use the general matrix smoothness assumption, i.e., that f_j is \mathbf{M}_j smooth. We recover results from [27]. Denote \mathbf{p} to be probability vector, i.e., $\mathbf{p}_i = \mathbb{P}(i \in R)$ where R is a random subset of $[n]$.

We shall consider the following choice of random operators \mathcal{S}, \mathcal{U} :

$$(\forall R) \text{ with probability } \mathbf{p}_R : \quad \mathcal{S}\mathbf{X} = \mathbf{X} \sum_{j \in R} e_j e_j^\top \quad \text{and} \quad \mathcal{U}\mathbf{X} = \mathbf{X} \sum_{j \in R} \frac{1}{\mathbf{p}_j} e_j e_j^\top.$$

The resulting algorithm is stated as Algorithm 3.

Algorithm 3 SAGA with arbitrary sampling (a variant of [27])

Require: learning rate $\alpha > 0$, starting point $x^0 \in \mathbb{R}^d$, random sampling $R \subseteq \{1, 2, \dots, n\}$

Set $\phi_j^0 = x^0$ for each $j \in [n]$

for $k = 0, 1, 2, \dots$ **do**

 Sample random $R^k \subseteq \{1, 2, \dots, n\}$

$$\text{Set } \phi_j^{k+1} = \begin{cases} x^k & j \in R^k \\ \phi_j^k & j \notin R^k \end{cases}$$

$$g^k = \frac{1}{n} \sum_{j=1}^n \nabla f_j(\phi_j^k) + \sum_{j \in R^k} \frac{1}{n\mathbf{p}_j} (\nabla f_j(\phi_j^{k+1}) - \nabla f_j(\phi_j^k))$$

$$x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$$

end for

In order to give tight rates under \mathbf{M} -smoothness, we need to do a bit more work. First, let $v \in \mathbb{R}^n$ be a vector for which the following inequality *expected separable overapproximation* inequality holds

$$\mathbb{E} \left[\left\| \sum_{j \in R} \mathbf{M}_j^{\frac{1}{2}} h_j \right\|^2 \right] \leq \sum_{j=1}^n p_j v_j \|h_j\|^2, \quad \forall h_1, \dots, h_n \in \mathbb{R}^d. \quad (32)$$

Since the function on the left is a quadratic in $h = (h_1, \dots, h_n) \in \mathbb{R}^{nd}$, this inequality is satisfied for large enough values of v_j . A variant of (32) was used to obtain the best known rates for coordinate descent with arbitrary sampling [28, 29].

Further, we shall consider the following assumption:

Assumption G.2. *Suppose that for all k*

$$\mathbf{G}(x^k) - \mathbf{G}(x^*) = \mathcal{M}^\dagger \mathcal{M} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \quad (33)$$

and

$$\mathbf{J}^k - \mathbf{G}(x^*) = \mathcal{M}^\dagger \mathcal{M} (\mathbf{J}^k - \mathbf{G}(x^*)). \quad (34)$$

The assumption, although in a slightly less general form, was demonstrated to obtain tightest complexity results for SAGA [27]. Note that if for each j , f_j corresponds to loss function of a linear model, then (33) and (34) follow for free. Further, Lemmas E.3 and E.4 give some easy-to-interpret sufficient conditions, such as lower boundedness of all functions f_j (which happens for any loss function), or twice differentiability of all functions f_j .

Corollary G.3 (Convergence rate of SAGA). *Let $\alpha = \min_j \frac{np_j}{4v_j + n\sigma}$. Then the iteration complexity of Algorithm 3 is $\max_j \left(\frac{4v_j + n\sigma}{n\sigma p_j} \right) \log \frac{1}{\epsilon}$.*

Remark G.4. *Corollary G.3 is slightly more general than Theorem 4.6 from [27] does not explicitly require linear models and \mathbf{M} smoothness implied by the linearity.*

H Special Cases: SEGA-like Methods

Let $n = 1$. Note that now operators \mathcal{S} and \mathcal{U} act on $d \times n$ matrices, i.e., on vectors in \mathbb{R}^d . To simplify notation, instead of $\mathbf{X} \in \mathbb{R}^{d \times n}$ we will write $x = (x_1, \dots, x_d) \in \mathbb{R}^d$.

H.1 Basic variant of SEGA [10]

Suppose that f is m -smooth (i.e., $\mathbf{M}_1 = m\mathbf{I}_d$) with $m > 0$. To recover basic SEGA from [10], consider the following choice of random operators \mathcal{S} and \mathcal{U} :

$$(\forall i) \text{ with probability } \frac{1}{d}: \quad \mathcal{S}x = \mathbf{e}_i \mathbf{e}_i^\top x = x_i \mathbf{e}_i \quad \text{and} \quad \mathcal{U}x = d \mathbf{e}_i \mathbf{e}_i^\top x = dx_i \mathbf{e}_i.$$

The resulting algorithm is stated as Algorithm 4.

Corollary H.1 (Convergence rate of SEGA). *Let $\alpha = \frac{1}{4md + \sigma d}$. Then the iteration complexity of Algorithm 4 is $(4 \frac{md}{\sigma} + d) \log \frac{1}{\epsilon}$.*

H.2 SEGA with arbitrary sampling

Consider a more general setup to that in Section H.1 and let us allow the smoothness matrix to be an arbitrary diagonal (positive semidefinite) matrix: $\mathbf{M} = \mathbf{D}(m_1, \dots, m_d)$ with $m_1, \dots, m_d > 0$. In this regime, we will establish a convergence rate for an arbitrary sampling strategy, and then use this to develop importance sampling.

Algorithm 4 SEGA [10]

Require: Stepsize $\alpha > 0$, starting point $x^0 \in \mathbb{R}^d$
 Set $h^0 = 0$
for $k = 0, 1, 2, \dots$ **do**
 Sample $i \in \{1, 2, \dots, d\}$ uniformly at random
 Set $h^{k+1} = h^k + (\nabla_i f(x^k) - h_i^k) \mathbf{e}_i$
 $g^k = h^k + d(\nabla_i f(x^k) - h_i^k) \mathbf{e}_i$
 $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$
end for

Let $\mathbf{p} \in \mathbb{R}^d$ be a probability vector with entries $\mathbf{p}_i = \mathbb{P}(i \in L)$. Consider the following choice of random operators \mathcal{S} and \mathcal{U} :

$$(\forall L) \text{ with prob. } \mathbf{p}_L : \mathcal{S}x = \sum_{i \in L} \mathbf{e}_i \mathbf{e}_i^\top x = \sum_{i \in L} x_i \mathbf{e}_i \quad \text{and} \quad \mathcal{U}x = \sum_{i \in L} \frac{1}{\mathbf{p}_i} \mathbf{e}_i \mathbf{e}_i^\top x = \sum_{i \in L} \frac{x_i}{\mathbf{p}_i} \mathbf{e}_i. \quad (35)$$

The resulting algorithm is stated as Algorithm 5.

Algorithm 5 SEGA with arbitrary sampling

Require: Stepsize $\alpha > 0$, starting point $x^0 \in \mathbb{R}^d$, random sampling $L \subseteq \{1, 2, \dots, d\}$
 Set $h^0 = 0$
for $k = 0, 1, 2, \dots$ **do**
 Sample random $L^k \subseteq \{1, 2, \dots, d\}$
 Set $h^{k+1} = h^k + \sum_{i \in L^k} (\nabla_i f(x^k) - h_i^k) \mathbf{e}_i$
 $g^k = h^k + \sum_{i \in L^k} \frac{1}{\mathbf{p}_i} (\nabla_i f(x^k) - h_i^k) \mathbf{e}_i$
 $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$
end for

Corollary H.2 (Convergence rate of SEGA). *Iteration complexity of Algorithm 5 with $\alpha = \min_i \frac{\mathbf{p}_i}{4m_i + \sigma}$ is $\max_i \left(\frac{4m_i + \sigma}{\mathbf{p}_i \sigma} \right) \log \frac{1}{\epsilon}$.*

Corollary H.2 indicates an up to constant factor optimal choice $\mathbf{p}_i \propto m_i$, which yields, up to a constant factor, $\frac{\sum_{i=1}^d m_i}{\sigma} \log \frac{1}{\epsilon}$ complexity. In the applications where m is not unique¹⁰, it is the best to choose one which minimizes $m^\top \mathbf{e}$.

Remark H.3. *Note that if $\mathbf{p}_i = 1$ for all i (i.e., if $\mathcal{U} = \mathcal{I}$), we recover proximal gradient descent as a special case.*

H.3 SVRCD with arbitrary sampling

As as a particular special case of Algorithm 1 we get a new method, which we call *Stochastic Variance Reduced Coordinate Descent* (SVRCD). The algorithm is similar to SEGA. The main difference is that SVRCD does not *update* a subset L of coordinates of vector h^k each iteration. Instead, with probability ρ , it *sets* h^k to $\nabla f(x^k)$.

¹⁰For example when a general matrix smoothness holds; one has to upper bound it by a diagonal matrix in order to comply with the assumptions of the section. In such case, there is an infinite array of possible choices of m .

We choose \mathcal{S} and \mathcal{U} via

$$\mathcal{S}\mathbf{X} = \begin{cases} 0 & \text{w.p. } 1 - \rho \\ \mathbf{X} & \text{w.p. } \rho \end{cases} \quad \text{and} \quad (\forall L) \quad \text{w.p. } \rho_L : \mathcal{U}\mathbf{X} = \sum_{i \in L} \frac{1}{\rho_i} \mathbf{e}_i \mathbf{e}_i^\top \mathbf{X},$$

where again $\rho_i = \mathbb{P}(i \in L)$. The randomness of \mathcal{S} is independent from the randomness of \mathcal{U} (which comes from the randomness of L). The resulting algorithm is stated as Algorithm 6.

Algorithm 6 SVRCD [NEW METHOD]

Require: starting point $x^0 \in \mathbb{R}^d$, random sampling $L \subseteq \{1, 2, \dots, d\}$, probability ρ , stepsize $\alpha > 0$
Set $h^0 = 0$
for $k = 0, 1, 2, \dots$ **do**
 Sample random $L^k \subseteq \{1, 2, \dots, d\}$
 $g^k = h^k + \sum_{i \in L^k} \frac{1}{\rho_i} (\nabla_i f(x^k) - h_i^k) \mathbf{e}_i$
 $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$
 Set $h^{k+1} = \begin{cases} h^k & \text{with probability } 1 - \rho \\ \nabla f(x^k) & \text{with probability } \rho \end{cases}$
end for

As in Section Q, we shall assume that f is $\mathbf{M} = \mathbf{D}(m_1, \dots, m_d)$ -smooth.

Corollary H.4. *Iteration complexity of Algorithm 6 with $\alpha = \min_i \frac{1}{4m_i/\rho_i + \sigma/\rho}$ is $\left(\frac{1}{\rho} + \max_i \frac{4m_i}{\rho_i \sigma}\right) \log \frac{1}{\epsilon}$.*

Corollary H.4 indicates optimal choice $p \propto m$.

Remark H.5. *If $\rho_i = 1$ for all i and $\rho = 1$, we recover proximal gradient descent as a special case.*

I Special Cases: SGD-shift

Suppose that $\mathbf{G}(x^*)$ is known. We will show that a version of SGD-AS converges with linear rate in such case. Let $\mathbf{J}^0 = \mathbf{G}(x^*)$. Consider the following choice of random operators \mathcal{S}, \mathcal{U} :

$$\mathcal{S}\mathbf{X} = 0 \quad \text{and} \quad (\forall R) \quad \text{with probability } \rho_R : \quad \mathcal{U}\mathbf{X} = \mathbf{X} \sum_{j \in R} \frac{1}{\rho_j} \mathbf{e}_j \mathbf{e}_j^\top.$$

The resulting algorithm is stated as Algorithm 7.

Algorithm 7 SGD-shift [NEW METHOD]

Require: learning rate $\alpha > 0$, starting point $x^0 \in \mathbb{R}^d$, random sampling $R \subseteq \{1, 2, \dots, n\}$
for $k = 0, 1, 2, \dots$ **do**
 Sample random $R^k \subseteq \{1, 2, \dots, n\}$
 $g^k = \frac{1}{n} \mathbf{G}(x^*) \mathbf{e} + \sum_{j \in R^k} \frac{1}{n\rho_j} (\nabla f_j(x^k) - \nabla f_j(x^*))$
 $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$
end for

Corollary I.1 (Convergence rate of SGD-AS-shift). *Suppose that f_j is \mathbf{M}_j -smooth for all j and suppose that v satisfies (32). Let $\alpha = n \min_j \frac{\rho_j}{v_j}$. Then, iteration complexity of Algorithm 7 is $\max_j \left(\frac{v_j}{n\rho_j \sigma}\right) \log \frac{1}{\epsilon}$.*

Remark I.2. *In overparameterized models, one has $\mathbf{G}(x^*) = 0$. In such a case, Algorithm 7 becomes SGD-AS [5], and we recover its tight convergence rate.*

J Special Cases: Loopless SVRG with Arbitrary Sampling (LSVRG)

In this section we extend Loopless SVRG (i.e., LSVRG) from [12, 15] to arbitrary sampling.

The main difference to SAGA is that LSVRG does not update \mathbf{J}^k at all with probability $1 - \rho$. However, with probability $1 - \rho$, it sets \mathbf{J}^k to $\mathbf{G}(x^k)$.

Define \mathcal{S} and \mathcal{U} as follows:

$$\mathcal{S}\mathbf{X} = \begin{cases} 0 & \text{w.p. } 1 - \rho \\ \mathbf{X} & \text{w.p. } \rho \end{cases} \quad \text{and} \quad (\forall R) \text{ with probability } p_R : \mathcal{U}\mathbf{X} = \mathbf{X} \sum_{i \in R} \frac{1}{p_j} e_j e_j^\top,$$

where $p_j = \mathbb{P}(j \in R)$.

The resulting algorithm is stated as Algorithm 8.

Algorithm 8 LSVRG (LSVRG [15] with arbitrary sampling) [NEW METHOD]

Require: learning rate $\alpha > 0$, starting point $x^0 \in \mathbb{R}^d$, random sampling $R \subseteq \{1, 2, \dots, n\}$

Set $\phi = x^0$

for $k = 0, 1, 2, \dots$ **do**

 Sample a random subset $R^k \subseteq \{1, 2, \dots, n\}$

$$g^k = \frac{1}{n} \sum_{j=1}^n \nabla f_j(\phi^k) + \sum_{j \in R^k} \frac{1}{np_j} (\nabla f_j(x^k) - \nabla f_j(\phi^k))$$

$$x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$$

$$\text{Set } \phi^{k+1} = \begin{cases} x^k & \text{with probability } \rho \\ \phi^k & \text{with probability } 1 - \rho \end{cases}$$

end for

In order to give tight rates under \mathbf{M} -smoothness, we shall consider ESO assumption (32) and Assumption G.2 (same as for SAGA-AS).

The next corollary shows the convergence result.

Corollary J.1 (Convergence rate of LSVRG). *Let $\alpha = \min_j \frac{n}{4 \frac{v_j}{p_j} + \frac{\sigma n}{\rho}}$. Then, iteration complexity of*

Algorithm 8 is $\max_j \left(4 \frac{v_j}{n\sigma p_j} + \frac{1}{\rho} \right) \log \frac{1}{\epsilon}$.

Remark J.2. *One can consider a slightly more general setting with $\mathcal{S}\mathbf{X} = \begin{cases} 0 & \text{w.p. } 1 - \rho \\ \mathbf{X} \sum_{i \in R'} e_j e_j^\top & \text{w.p. } \rho \end{cases}$, where distribution of $R' \subseteq [n]$ is arbitrary. Clearly, such methods is a special case of Algorithm 1, and setting $R' = [n]$ with probability 1, LSVRG is obtained. However, in a general form, such algorithm resembles SCSG [18]. However, unlike SCSG, the described method converges linearly, thus is superior to SCSG.*

K Special Cases: Methods with Bernoulli \mathcal{U}

Throughout this section, we will suppose that $\mathbf{M}_j = m\mathbf{I}_d$ for all j . This is sufficient to establish strong results. Indeed, Bernoulli \mathcal{U} does not allow for an efficient importance sampling and hence one can't develop arbitrary sampling results similar to those in Section G.2 or Section Q.

K.1 B2 (Bernoulli \mathcal{S})

Let $n = 1$. Note that now operators \mathcal{S} and \mathcal{U} act on $d \times n$ matrices, i.e., on vectors in \mathbb{R}^d . To simplify notation, instead of $\mathbf{X} \in \mathbb{R}^{d \times n}$ we will write $x = (x_1, \dots, x_d) \in \mathbb{R}^d$. Given probabilities

$0 < \rho, \delta \leq 1$, let both \mathcal{S} and \mathcal{U} be Bernoulli (i.e., scaling) sketches:

$$\mathcal{S}x = \begin{cases} 0 & \text{w.p. } 1 - \rho \\ x & \text{w.p. } \rho \end{cases} \quad \text{and} \quad \mathcal{U}x = \begin{cases} 0 & \text{w.p. } 1 - \delta \\ \frac{1}{\delta}x & \text{w.p. } \delta \end{cases}.$$

The resulting algorithm is stated as Algorithm 9.

Algorithm 9 B2 [NEW METHOD]

Require: learning rate $\alpha > 0$, starting point $x^0 \in \mathbb{R}^d$, probabilities $\delta \in (0, 1]$ and $\rho \in (0, 1]$

Set $\phi = x^0$

for $k = 0, 1, 2, \dots$ **do**

$$g^k = \begin{cases} \nabla f(\phi^k) & \text{w.p. } 1 - \delta \\ \frac{1}{\delta} \nabla f(x^k) - \left(\frac{1}{\delta} - 1\right) \nabla f(\phi^k) & \text{w.p. } \delta \end{cases}$$

$$x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$$

$$\text{Set } \phi^{k+1} = \begin{cases} x^k & \text{w.p. } \rho \\ \phi^k & \text{w.p. } 1 - \rho \end{cases}$$

end for

Corollary K.1 (Convergence rate B2). *Suppose that f is m -smooth. Let $\alpha = \frac{1}{4\frac{m}{\delta} + \frac{\sigma}{\rho}}$. Then, iteration complexity of Algorithm 9 is $\left(4\frac{m}{\sigma\delta} + \frac{1}{\rho}\right) \log \frac{1}{\epsilon}$.*

Remark K.2. *It is possible to choose correlated \mathcal{S} and \mathcal{U} without any sacrifice in the rate.*

K.2 LSVRG-inv (Right \mathcal{S})

Given a probability scalar $0 < \delta \leq 1$, consider choosing operators \mathcal{S} and \mathcal{U} as follows:

$$\mathcal{S}\mathbf{X} = \mathbf{X} \sum_{j \in R} e_j e_j^\top \quad \text{w.p. } p_R \quad \text{and} \quad \mathcal{U}\mathbf{X} = \begin{cases} 0 & \text{w.p. } 1 - \delta \\ \frac{1}{\delta} \mathbf{X} & \text{w.p. } \delta. \end{cases}$$

The resulting algorithm is stated as Algorithm 10.

Algorithm 10 LSVRG-inv [NEW METHOD]

Require: starting point $x^0 \in \mathbb{R}^d$, random sampling $R \subseteq \{1, 2, \dots, n\}$, probability $\delta \in (0, 1]$, learning rate $\alpha > 0$

Set $\phi_j^0 = x^0$ for $j = 1, 2, \dots, n$

for $k = 0, 1, 2, \dots$ **do**

$$g^k = \begin{cases} \frac{1}{n} \sum_{j=1}^n \nabla f_j(\phi_j^k) & \text{w.p. } 1 - \delta \\ \frac{1}{\delta} \nabla f(x^k) - \left(\frac{1}{\delta} - 1\right) \frac{1}{n} \sum_{j=1}^n \nabla f_j(\phi_j^k) & \text{w.p. } \delta \end{cases}$$

$$x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$$

Sample a random subset $R^k \subseteq \{1, 2, \dots, n\}$

$$\text{Set } \phi_j^{k+1} = \begin{cases} x^k & j \in R^k \\ \phi_j^k & j \notin R^k \end{cases}$$

end for

Corollary K.3 (Convergence rate of LSVRG-inv). *Suppose that each f_i is m -smooth. Let $\alpha = \min_j \frac{1}{4\frac{m}{\delta} + \frac{\sigma}{p_j}}$. Then, iteration complexity of Algorithm 10 is $\max_j \left(4\frac{m}{\sigma\delta} + \frac{1}{p_j}\right) \log \frac{1}{\epsilon}$.*

K.3 SVRCD-inv (Left \mathcal{S})

Let $n = 1$. Note that now operators \mathcal{S} and \mathcal{U} act on $d \times n$ matrices, i.e., on vectors in \mathbb{R}^d . To simplify notation, instead of $\mathbf{X} \in \mathbb{R}^{d \times n}$ we will write $x = (x_1, \dots, x_d) \in \mathbb{R}^d$.

Consider again setup where $n = 1$. Choose operators \mathcal{S} and \mathcal{U} as follows:

$$\mathcal{S}x = \sum_{i \in L} \mathbf{e}_i \mathbf{e}_i^\top x \quad \text{w.p. } p_L \quad \text{and} \quad \mathcal{U}x = \begin{cases} 0 & \text{w.p. } 1 - \delta \\ \frac{1}{\delta} x & \text{w.p. } \delta. \end{cases}$$

For convenience, let \mathbf{p} be the probability vector defined as: $p_i = \mathbb{P}(i \in L)$.

The resulting algorithm is stated as Algorithm 11.

Algorithm 11 SVRCD-inv [NEW METHOD]

Require: starting point $x^0 \in \mathbb{R}^d$, random sampling $L \subseteq \{1, 2, \dots, d\}$, probability $\delta \in (0, 1]$, learning rate $\alpha > 0$

Choose $h^0 \in \mathbb{R}^d$

for $k = 0, 1, 2, \dots$ **do**

$$g^k = \begin{cases} h^k & \text{w.p. } 1 - \delta \\ \frac{1}{\delta} \nabla f(x^k) - \left(\frac{1}{\delta} - 1\right) h^k & \text{w.p. } \delta \end{cases}$$

$$x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$$

Sample a random subset $L^k \subseteq \{1, 2, \dots, d\}$

$$\text{Set } h^{k+1} = h^k + \sum_{i \in L^k} (\nabla_i f(x^k) - h_i^k) \mathbf{e}_i$$

end for

Corollary K.4 (Convergence rate of SVRCD-inv). *Suppose that each f_j is m -smooth. Let $\alpha = \min_i \frac{1}{4\frac{m}{\delta} + \frac{\sigma}{p_i}}$. Then, iteration complexity of Algorithm 11 is $\max_i \left(4\frac{m}{\delta} + \frac{1}{p_i}\right) \log \frac{1}{\epsilon}$.*

L Special Cases: Combination of Left and Right Sketches

L.1 RL (right sampling \mathcal{S} , left unbiased sampling \mathcal{U})

Consider choosing \mathcal{S} and \mathcal{U} as follows:

$$\mathcal{S}\mathbf{X} = \mathbf{X} \sum_{j \in R} \mathbf{e}_j \mathbf{e}_j^\top \quad \text{w.p. } p_R \quad \text{and} \quad \mathcal{U}\mathbf{X} = \sum_{i \in L} \frac{1}{p_i} \mathbf{e}_i \mathbf{e}_i^\top \mathbf{X} \quad \text{w.p. } p_L.$$

The resulting algorithm is stated as Algorithm 12.

Corollary L.1 (Convergence rate of RL). *Suppose that each f_j is $\mathbf{D}(m^j)$ -smooth, where $m^j \in \mathbb{R}^d$ and $\mathbf{D}(m^j) \succ 0$. Let $\alpha = \min_{i,j} \left(4\frac{m_i^j}{p_i} + \frac{\sigma}{p_j}\right)^{-1}$. Then, iteration complexity of Algorithm 12 is $\max_{i,j} \left(4\frac{m_i^j}{\sigma p_i} + \frac{1}{p_j}\right) \log \frac{1}{\epsilon}$.*

L.2 LR (left sampling \mathcal{S} , right unbiased sampling \mathcal{U})

Consider choosing \mathcal{S} and \mathcal{U} as follows:

$$\mathcal{S}\mathbf{X} = \sum_{i \in L} \mathbf{e}_i \mathbf{e}_i^\top \mathbf{X} \quad \text{w.p. } p_L \quad \text{and} \quad \mathcal{U}\mathbf{X} = \mathbf{X} \sum_{j \in R} \frac{1}{p_j} \mathbf{e}_j \mathbf{e}_j^\top \quad \text{w.p. } p_R.$$

The resulting algorithm is stated as Algorithm 13.

Algorithm 12 RL [NEW METHOD]

Require: starting point $x^0 \in \mathbb{R}^d$, random sampling $L \subseteq \{1, 2, \dots, d\}$, random sampling $R \subseteq \{1, 2, \dots, n\}$, learning rate $\alpha > 0$
Set $\phi_j^0 = x^0$ for each j
for $k = 0, 1, 2, \dots$ **do**
 Sample random $R^k \subseteq \{1, 2, \dots, n\}$
 Set $\phi_j^{k+1} = \begin{cases} x^k & j \in R^k \\ \phi_j^k & j \notin R^k \end{cases}$
 Sample random $L^k \subseteq \{1, 2, \dots, d\}$
 $g^k = \frac{1}{n} \sum_{j=1}^n \nabla f_j(\phi_j^k) + \sum_{i \in L^k} \frac{1}{\mathbf{p}_i} \left(\nabla_i f(x^k) - \frac{1}{n} \sum_{j=1}^n \nabla_i f_j(\phi_j^k) \right) \mathbf{e}_i$
 $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$
end for

Algorithm 13 LR [NEW METHOD]

Require: starting point $x^0 \in \mathbb{R}^d$, random sampling $L \subseteq \{1, 2, \dots, d\}$, random sampling $R \subseteq \{1, 2, \dots, n\}$, learning rate $\alpha > 0$
Set $h^0 = x^0$ for each j
for $k = 0, 1, 2, \dots$ **do**
 Sample random $L^k \subseteq \{1, 2, \dots, d\}$
 Set $h^{k+1} = h^k + \sum_{i \in L^k} (\nabla_i f(x^k) - h_i^k) \mathbf{e}_i$
 Sample random $R^k \subseteq \{1, 2, \dots, n\}$
 $g^k = \nabla f(h^k) + \sum_{j \in R^k} \frac{1}{n\mathbf{p}_j} (\nabla f_j(x^k) - \nabla f_j(h^k))$
 $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$
end for

Corollary L.2 (Convergence rate of LR). *Suppose that each f_j is \mathbf{M}_j -smooth, and suppose that $v \in \mathbb{R}^n$ is such that (32) holds. Let $\alpha = \min_{i,j} \frac{1}{4v_j\mathbf{p}_j^{-1} + \sigma\mathbf{p}_i^{-1}}$. Then, iteration complexity of Algorithm 13 is $\max_{i,j} \left(4\frac{v_i}{\sigma\mathbf{p}_j} + \frac{1}{\mathbf{p}_i} \right) \log \frac{1}{\epsilon}$.*

M Special Cases: Joint Left and Right Sketches

M.1 SAEGA

Another new special case of Algorithm 1 we propose is SAEGA (the name comes from the combination of names SAGA and SEGA). In SAEGA, both \mathcal{S} and \mathcal{U} are fully correlated and consist of right and left sketch. However, the mentioned right and left sketches are independent. In particular, we have

$$\mathcal{S}\mathbf{X} = \mathbf{X}_{LR} = \left(\sum_{i \in L} \mathbf{e}_i \mathbf{e}_i^\top \right) \mathbf{X} \left(\sum_{j \in R} \mathbf{e}_j \mathbf{e}_j^\top \right), \quad L \subset [d], R \subset [n] \quad \text{are independent random sets.}$$

Next, \mathcal{U} is chosen as

$$\mathcal{U}\mathbf{X} = \mathcal{S} \left(\left(\mathbf{p}^{-1} (\mathbf{p}^{-1})^\top \right) \circ \mathbf{X} \right)$$

where $\mathbf{p}_i = \mathbb{P}(i \in L)$ and $\mathbf{p}_j = \mathbb{P}(j \in R)$. The resulting algorithm is stated as Algorithm 14.

Suppose that for all $j \in [n]$, $\mathbf{M}_j = \mathbf{D}(m^j) \succ 0$ is diagonal matrix¹¹.

¹¹A block diagonal matrix \mathbf{M}_j with blocks such that $\mathcal{M}\mathcal{P}_\mathcal{S} = \mathcal{P}_\mathcal{S}\mathcal{M}$ would work as well

Algorithm 14 SAEGA [NEW METHOD]

Input: $x^0 \in \mathbb{R}^d$, random sampling $L \subseteq \{1, 2, \dots, d\}$, random sampling $R \subseteq \{1, 2, \dots, n\}$, stepsize α
 $\mathbf{J}^0 = 0$
for $k = 0, 1, \dots$ **do**
 Sample random $L^k \subseteq \{1, 2, \dots, d\}$ and $R^k \subseteq \{1, 2, \dots, n\}$
 Compute $\nabla_i f_j(x^k)$ for all $i \in L^k$ and $j \in R^k$
 $\mathbf{J}_{ij}^{k+1} = \begin{cases} \nabla_i f_j(x^k) & i \in L^k \text{ and } j \in R^k \\ \mathbf{J}_{ij}^k & \text{otherwise} \end{cases}$
 $g^k = \left(\mathbf{J}^k + \left(\textcolor{red}{p}^{-1} (\textcolor{blue}{p}^{-1})^\top \right) \circ (\mathbf{J}^{k+1} - \mathbf{J}^k) \right) \textcolor{blue}{e}$
 $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$
end for

Let $\mathbf{P} \in \mathbb{R}^{n \times n}$ be the probability matrix with respect to R -sampling, i.e., $\mathbf{P}_{jj'} = \mathbb{P}(j \in R, j' \in R)$.

Corollary M.1. Consider any (elementwise) positive vector $\textcolor{blue}{q}$ such that $\mathbf{D}(\textcolor{blue}{p})^{-1} \mathbf{P} \mathbf{D}(\textcolor{blue}{p})^{-1} \preceq \mathbf{D}(\textcolor{blue}{q})^{-1}$. Let $\alpha = \min_{i,j} \frac{n \textcolor{red}{p}_i \textcolor{blue}{q}_j}{4m_i^j + n\sigma}$. Then, iteration complexity of Algorithm 14 is $\max_{i,j} \left(4 \frac{m_i^j}{\sigma n \textcolor{red}{p}_i \textcolor{blue}{q}_j} + \frac{1}{\textcolor{red}{p}_i} \frac{1}{\textcolor{blue}{q}_j} \right) \log \frac{1}{\epsilon}$.

M.2 SVRC DG

Next new special case of Algorithm 1 we propose is SVRC DG. SVRC DG uses the same random operator \mathcal{U} as SAEGA. The difference to SAEGA lies in operator \mathcal{S} which is Bernoulli random variable:

$$\mathcal{S}\mathbf{X} = \begin{cases} 0 & \text{w.p. } 1 - \textcolor{blue}{\rho} \\ \mathbf{X} & \text{w.p. } \textcolor{blue}{\rho} \end{cases}, \quad \mathcal{U}\mathbf{X} = \mathbf{I}_L: \left(\left(\textcolor{red}{p}^{-1} (\textcolor{blue}{p}^{-1})^\top \right) \circ \mathbf{X} \right) \mathbf{I}_{:R},$$

where $L \subseteq [d]$, and $R \subseteq [n]$ are independent random sets and $\textcolor{red}{p}_i = \mathbb{P}(i \in L)$ and $\textcolor{blue}{p}_j = \mathbb{P}(j \in R)$.

The resulting algorithm is stated as Algorithm 15.

Algorithm 15 SVRC DG [NEW METHOD]

Input: $x^0 \in \mathbb{R}^d$, random sampling $L \subseteq \{1, 2, \dots, d\}$, random sampling $R \subseteq \{1, 2, \dots, n\}$, stepsize α , probability $\textcolor{blue}{\rho}$
 $\mathbf{J}^0 = 0$
for $k = 0, 1, \dots$ **do**
 Sample random $L^k \subseteq \{1, 2, \dots, d\}$ and $R^k \subseteq \{1, 2, \dots, n\}$
 Observe $\nabla_i f_j(x^k)$ for all $i \in L^k$ and $j \in R^k$
 $g^k = \left(\mathbf{J}^k + \left(\textcolor{red}{p}^{-1} (\textcolor{blue}{p}^{-1})^\top \right) \circ (\mathbf{I}_{L^k:} (\mathbf{G}(x^k) - \mathbf{J}^k) \mathbf{I}_{:R^k}) \right) \textcolor{blue}{e}$
 $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$
 $\mathbf{J}^{k+1} = \begin{cases} \mathbf{G}(x^k) & \text{with probability } \textcolor{blue}{\rho} \\ \mathbf{J}^k & \text{with probability } 1 - \textcolor{blue}{\rho} \end{cases}$
end for

Suppose that for all j , $\mathbf{M}_j = \mathbf{D}(m^j)$ is diagonal matrix¹².

For notational simplicity, denote $\mathbf{M}' \in \mathbb{R}^{d \times n}$ to be the matrix with j -th column equal to m_j .

Let $\mathbf{P} \in \mathbb{R}^{n \times n}$ be the probability matrix with respect to R -sampling, i.e., $\mathbf{P}_{jj'} = \mathbb{P}(j \in R, j' \in R)$.

¹²Block diagonal \mathbf{M}_j with blocks such that $\mathcal{M}\mathcal{S} = \mathcal{S}\mathcal{M}$ would work as well

Corollary M.2. Consider any (elementwise) positive vector q such that $\mathbf{D}(p)^{-1}\mathbf{P}\mathbf{D}(p)^{-1} \preceq \mathbf{D}(q)^{-1}$. Let $\alpha = \min_{i,j} \frac{1}{4\frac{m_i^j}{p_i q_j n} + \frac{1}{\rho}\sigma}$. Then, iteration complexity of Algorithm 15 is $\max_{i,j} \left(4\frac{m_i^j}{\sigma n p_i q_j} + \frac{1}{\rho}\right) \log \frac{1}{\epsilon}$.

M.3 ISAEGA (with distributed data)

In this section, we consider a distributed setting from [21]. In particular, [21] proposed a strategy of running coordinate descent on top of various optimization algorithms such as GD, SGD or SAGA, while keeping the convergence rate of the original method. This allows for sparse communication from workers to master.

However, ISAGA (distributed SAGA with RCD on top of it), as proposed, assumes zero gradients at the optimum which only holds for overparameterized models. It was stated as an open question whether it is possible to derive SEGA on top of it such that the mentioned assumption can be dropped. We answer this question positively, proposing ISAEGA (Algorithm 16). Next, algorithms proposed in [21] only allow for uniform sampling under simple smoothness. In contrast, we develop an arbitrary sampling strategy for general matrix smoothness¹³.

Assume that we have T parallel units, each owning set of indices N_t (for $1 \leq t \leq T$). Next, consider distributions \mathcal{D}_t over subsets of N_t and distributions \mathcal{D}_t over subsets coordinates $[d]$ for each machine. Each iteration we sample $R_t \sim \mathcal{D}_t, L_t \sim \mathcal{D}_t$ (for $1 \leq t \leq T$) and observe the corresponding part of Jacobian $\mathbf{J}_{\cap_t(L_t, R_t)}^k$. Thus the corresponding random Jacobian sketch becomes

$$\mathbf{S}\mathbf{X} = \mathbf{X}_{\cap_t(L_t, R_t)} = \sum_{t=1}^T \left(\sum_{i \in L_t} \mathbf{e}_i \mathbf{e}_i^\top \right) \mathbf{X}_{:N_t} \left(\sum_{j \in R_t} \mathbf{e}_j \mathbf{e}_j^\top \right).$$

Next, for each $1 \leq t \leq T$ consider vector $\mathbf{p}^t \in \mathbb{R}^d, \mathbf{p}^t \in \mathbb{R}^{|N_t|}$ such that $\mathbb{P}(i \in L_t) = p_i^t$ and $\mathbb{P}(j \in R_t) = p_j^t$. Given the notation, random operator \mathcal{U} is chosen as

$$\mathcal{U}\mathbf{X} = \sum_{t=1}^T \left((\mathbf{p}^t)^{-1} ((\mathbf{p}^t)^{-1})^\top \right) \circ \left(\left(\sum_{i \in L_t} \mathbf{e}_i \mathbf{e}_i^\top \right) \mathbf{X}_{:N_t} \left(\sum_{j \in R_t} \mathbf{e}_j \mathbf{e}_j^\top \right) \right).$$

The resulting algorithm is stated as Algorithm 16.

Suppose that for all $1 \leq j \leq n$, $\mathbf{M}_j = \mathbf{D}(m_j)$ is diagonal matrix¹⁴. Let $\mathbf{P}^t \in \mathbb{R}^{\|N_t\| \times \|N_t\|}$ be the probability matrix with respect to R_t -sampling, i.e., $\mathbf{P}_{jj'}^t = \mathbb{P}(j \in R_t, j' \in R_t)$.

Corollary M.3. For all t consider any (elementwise) positive vector q^t such that $\mathbf{D}(\mathbf{p}^t)^{-1}\mathbf{P}^t\mathbf{D}(\mathbf{p}^t)^{-1} \preceq \mathbf{D}(q^t)^{-1}$. Let $\alpha = \min_{j \in N_t, i, t} \frac{1}{4m_i^j \left(1 + \frac{1}{n p_i^t q_j^t}\right) + \frac{\sigma}{p_i^t} q_j^t}$. Then, iteration complexity of Algorithm 16 is $\max_{j \in N_t, i, t} \left(4\frac{m_i^j}{\sigma} \left(1 + \frac{1}{n p_i^t q_j^t}\right) + \frac{1}{p_i^t q_j^t}\right) \log \frac{1}{\epsilon}$.

Thus, for all j , it does not make sense to increase sampling size beyond point where $p_i^t q_j^t \geq \frac{1}{n}$ as the convergence speed would not increase significantly¹⁵.

Remark M.4. In special case when $R_t = N_t$ always, ISAEGA becomes ISEGA from [21]. However [21] assumes that $|N_t|$ is constant in t and $L_t = \mathbf{e}_i$ with probability $\frac{1}{d}$. Thus, even special

¹³We do so only for ISAEGA. However, our framework allows obtaining arbitrary sampling results for ISAGA, ISEGA and ISGD (with no variance at optimum) as well. We omit it for space limitations

¹⁴block diagonal \mathbf{M}_j with blocks such that $\mathcal{M}\mathcal{S} = \mathcal{S}\mathcal{M}$ would work as well

¹⁵For indices i, j, t which maximize the rate from Corollary M.3.

Algorithm 16 ISAEGA [NEW METHOD]

Input: $x^0 \in \mathbb{R}^d$, # parallel units T , each owning set of indices N_t (for $1 \leq t \leq T$), distributions \mathcal{D}_t over subsets of N_t , distributions \mathcal{D}_t over subsets coordinates $[d]$, stepsize α
 $\mathbf{J}^0 = 0$
for $k = 0, 1, \dots$ **do**
 for $t = 1, \dots, T$ **in parallel do**
 Sample $R_t \sim \mathcal{D}_t$; $R_t \subseteq N_t$ (independently on each machine)
 Sample $L_t \sim \mathcal{D}_t$; $L_t \subseteq [d]$ (independently on each machine)
 Observe $\nabla_{L_t} f_j(x^k)$ for $j \in R_t$
 For $i \in [d], j \in N_t$ set $\mathbf{J}_{i,j}^{k+1} = \begin{cases} \nabla_i f_j(x^k) & \text{if } i \in [d], j \in R_t, i \in L_t \\ \mathbf{J}_{i,j}^k & \text{otherwise} \end{cases}$
 Send $\mathbf{J}_{:N_t}^{k+1} - \mathbf{J}_{:N_t}^k$ to master ▷ Sparse; low communication
 end for
 $g^k = \left(\mathbf{J}^k + \sum_{t=1}^T \left(p^{t-1} p^{t-1\top} \right) \circ \left(\left(\sum_{i \in L_t} e_i e_i^\top \right) (\mathbf{J}^{k+1} - \mathbf{J}^k)_{:N_t} \left(\sum_{j \in R_t} e_j e_j^\top \right) \right) \right) e$
 $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$
end for

case of Corollary M.3 generalizes results on ISEGA from [21]. For completeness, we state ISEGA as Algorithm 17 and Corollary M.5 provides its iteration complexity.

Algorithm 17 ISEGA (ISEGA [21] with arbitrary sampling) [NEW METHOD]

Input: $x^0 \in \mathbb{R}^d$, # parallel units T , each owning set of indices N_t (for $1 \leq t \leq T$), distributions \mathcal{D}_t over subsets coordinates $[d]$, stepsize α
 $\mathbf{J}^0 = 0$
for $k = 0, 1, \dots$ **do**
 for $t = 1, \dots, T$ **in parallel do**
 Sample $L_t \sim \mathcal{D}_t$; $L_t \subseteq [d]$ (independently on each machine)
 Observe $\nabla_{L_t} f_j(x^k)$ for $j \in N_t$
 For $i \in [d], j \in N_t$ set $\mathbf{J}_{i,j}^{k+1} = \begin{cases} \nabla_i f_j(x^k) & \text{if } i \in [d], j \in N_t, i \in L_t \\ \mathbf{J}_{i,j}^k & \text{otherwise} \end{cases}$
 Send $\mathbf{J}_{:N_t}^{k+1} - \mathbf{J}_{:N_t}^k$ to master ▷ Sparse; low communication
 end for
 $g^k = \left(\mathbf{J}^k + \sum_{t=1}^T \left(p^{t-1} e^\top \right) \circ \left(\left(\sum_{i \in L_t} e_i e_i^\top \right) (\mathbf{J}^{k+1} - \mathbf{J}^k)_{:N_t} \right) \right) e$
 $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$
end for

Corollary M.5. Let $\alpha = \min_{j \in N_t, i, t} \frac{1}{4m_i^j \left(1 + \frac{1}{np_i^t |N_t|} \right) + \frac{\sigma}{p_i^t} |N_t|}$. Then, iteration complexity of Algorithm 16 is $\max_{j \in N_t, i, t} \left(4 \frac{m_i^j}{\sigma} \left(1 + \frac{1}{np_i^t |N_t|} \right) + \frac{1}{p_i^t |N_t|} \right) \log \frac{1}{\epsilon}$.

N Special Cases: JacSketch

As next special case of GJS (Algorithm 1) we present JacSketch (JS) motivated by [8]. The algorithm observes every iteration a single right sketch of the Jacobian and constructs operators \mathcal{S}, \mathcal{U} in the following fashion:

$$\mathcal{S}\mathbf{X} = \mathbf{X}\mathbf{R} \quad \text{and} \quad \mathcal{U}\mathbf{X} = \mathbf{X}\mathbf{R}\mathbb{E}[\mathbf{R}]^{-1}$$

Algorithm 18 JS (JacSketch)

- 1: **Parameters:** Stepsize $\alpha > 0$, Distribution \mathcal{D} over random projector matrices $\mathbf{R} \in \mathbb{R}^{n \times n}$
 - 2: **Initialization:** Choose solution estimate $x^0 \in \mathbb{R}^d$ and Jacobian estimate $\mathbf{J}^0 \in \mathbb{R}^{d \times n}$
 - 3: **for** $k = 0, 1, \dots$ **do**
 - 4: Sample realization of $\mathbf{R} \sim \mathcal{D}$ perform sketches $\mathbf{G}(x^k)\mathbf{R}$
 - 5: $\mathbf{J}^{k+1} = \mathbf{J}^k - (\mathbf{J}^k - \mathbf{G}(x^k)\mathbf{R})$
 - 6: $g^k = \frac{1}{n}\mathbf{J}^k \mathbf{e} + \frac{1}{n}(\mathbf{G}(x^k) - \mathbf{J}^k) \mathbf{R} \mathbb{E}[\mathbf{R}]^{-1} \mathbf{e}$
 - 7: $x^{k+1} = \text{prox}_{\alpha\psi}(x^k - \alpha g^k)$
 - 8: **end for**
-

where $\mathbf{R} \in \mathbb{R}^{n \times n}$ is random projection matrix.

Note that Algorithm 18 differs to what was proposed in [8] in the following points.

- Approach from [8] uses a scalar random variable $\theta_{\mathbf{R}}$ to set $\mathcal{U}\mathbf{X} = \theta_{\mathbf{R}}\mathbf{X}\mathbf{R}$. Instead, we set $\mathbb{E}[\mathcal{U}] = \mathbf{X}\mathbf{R}\mathbb{E}[\mathbf{R}]^{-1}$. This tweak allows Algorithm 1 to recover the tightest known analysis of SAGA as a special case. Note that the approach from [8] only recovers tight rates for SAGA under uniform sampling.
- Unlike [8], our setup allows for proximable regularizer, thus is more general.
- Approach from [8] allows projections under a general weighted norm. Algorithm 1 only allows for non-weighted norm; which is only done for the sake of simplicity as the paper is already notation-heavy. However, GJS (Algorithm 1) is general enough to allow for an arbitrary weighted norm.

The next corollary shows the convergence result.

Corollary N.1 (Convergence rate of JacSketch). *Suppose that operator \mathcal{M} is commutative with right multiplication by \mathbf{R} always. Consider any $\mathbf{B} \in \mathbb{R}^{n \times n}$ which commutes with \mathbf{R} always. Denote*

$$\mathbf{M}^{\frac{1}{2}} := \begin{pmatrix} \mathbf{M}_1^{\frac{1}{2}} & & \\ & \ddots & \\ & & \mathbf{M}_n^{\frac{1}{2}} \end{pmatrix} \quad \text{and} \quad \eta := \lambda_{\max} \left(\mathbf{M}^{\frac{1}{2}^\top} \left(\mathbb{E} \left[\mathbf{R} \mathbb{E}[\mathbf{R}]^{-1} \mathbf{e} \mathbf{e}^\top \mathbb{E}[\mathbf{R}]^{-1} \mathbf{R} \right] \otimes \mathbf{I}_d \right) \mathbf{M}^{\frac{1}{2}} \right).$$

Let

$$\alpha = \frac{\lambda_{\min}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}] \mathbf{B})}{4n^{-1}\eta \lambda_{\max}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}] \mathbf{B}) + \sigma \lambda_{\max}(\mathbf{B}^\top \mathbf{B})}.$$

Then, iteration complexity of Algorithm 18 is

$$\frac{4n^{-1}\eta\sigma^{-1}\lambda_{\max}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}] \mathbf{B}) + \lambda_{\max}(\mathbf{B}^\top \mathbf{B})}{\lambda_{\min}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}] \mathbf{B})} \log \frac{1}{\epsilon}.$$

O Special Cases: Proofs

In this section, we provide the proofs of all corollaries listed in previous sections.

For simplicity, we will use the following notation throughout this section: $\Gamma(\mathbf{X}) = \mathcal{U}(\mathbf{X})e$.

O.1 SAGA methods: Proofs

O.1.1 Setup for Corollary G.1

Note first that the choice of \mathcal{S}, \mathcal{U} yields

$$\begin{aligned}\mathbb{E}[\mathcal{S}(\mathbf{X})] &= \frac{1}{n}\mathbf{X} \\ \mathbb{E}[\|\Gamma(\mathbf{X})\|^2] &= n^2\mathbb{E}[\langle \mathbf{X}^\top, \mathbf{e}_j \mathbf{e}_j^\top \mathbf{e} \mathbf{e}^\top \mathbf{e}_j \mathbf{e}_j^\top \mathbf{X}^\top \rangle] = n\|\mathbf{X}\|^2.\end{aligned}$$

Next, as we have no prior knowledge about $\mathbf{G}(x^*)$, let $\mathcal{R} \equiv \mathcal{I}$; i.e. $\text{Range}(\mathcal{R}) = \mathbb{R}^{d \times n}$. Lastly, consider \mathcal{B} operator to be a multiplication with constant β : $\mathcal{B}(\mathbf{X}) = \beta\mathbf{X}$.

Thus for (13) we should have

$$\frac{2\alpha}{n}m + \beta^2 \left(1 - \frac{1}{n}\right) \leq (1 - \alpha\sigma)\beta^2$$

and for (14) we should have

$$\frac{2\alpha}{n}m + \frac{\beta^2}{n} \leq \frac{1}{n}.$$

It remains to notice that choices $\alpha = \frac{1}{4m+\sigma n}$ and $\beta^2 = \frac{1}{2}$ are valid to satisfy the above bounds.

O.1.2 Setup for Corollary G.3

First note that $\mathbb{E}[\mathcal{S}(\mathbf{X})] = \mathbf{X}\mathbf{D}(p)$.

Next, due to (34), (33) with choice $\mathbf{Y} = \mathcal{M}^{\frac{1}{2}}\mathbf{X}$ (13) and (14) become respectively:

$$\frac{2\alpha}{n^2}\mathbb{E}\left[\left\|\sum_{j \in R} p_j^{-1} \mathbf{M}_j^{\frac{1}{2}} \mathbf{Y}_{:j}\right\|^2\right] + \left\|(\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}(\mathbf{Y})\right\|^2 \leq (1 - \alpha\sigma)\|\mathcal{B}(\mathbf{Y})\|^2 \quad (36)$$

$$\frac{2\alpha}{n^2}\mathbb{E}\left[\left\|\sum_{j \in R} p_j^{-1} \mathbf{M}_i^{\frac{1}{2}} \mathbf{Y}_{:i}\right\|^2\right] + \left\|(\mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}(\mathbf{Y})\right\|^2 \leq \frac{1}{n}\|\mathbf{Y}\|^2 \quad (37)$$

Note that

$$\mathbb{E}\left[\left\|\sum_{j \in R} p_j^{-1} \mathbf{M}_j^{\frac{1}{2}} \mathbf{Y}_{:j}\right\|^2\right] = \mathbb{E}\left[\left\|\sum_{j \in R} \mathbf{M}_j^{\frac{1}{2}} (p_j^{-1} \mathbf{Y}_{:j})\right\|^2\right] \leq \sum_{j=1}^n p_j^{-1} v_j \|\mathbf{Y}_{:j}\|^2$$

where we used ESO assumption (32) in the last bound above.

Next choose \mathcal{B} to be right multiplication with $\mathbf{D}(b)$. Thus, for (36) it suffices to have for all $j \in [n]$

$$\frac{2\alpha}{n^2} v_j p_j^{-1} + b_j^2 (1 - p_j) \leq b_j^2 (1 - \alpha\sigma) \quad \Rightarrow \quad \frac{2\alpha}{n^2} v_j p_j^{-1} + b_j^2 \alpha\sigma \leq b_j^2 p_j$$

For (37) it suffices to have for all $j \in [n]$

$$\frac{2\alpha}{n^2} v_j p_j^{-1} + b_j^2 p_j \leq \frac{1}{n}$$

It remains to notice that choice $b_j^2 = \frac{1}{2n p_j}$ and $\alpha = \min_j \frac{n p_j}{4v_j + n\sigma}$ is valid.

O.2 SEGA methods: Proofs

O.2.1 Setup for Corollary H.1

Note that

$$\begin{aligned}\mathbb{E}[\mathcal{S}x] &= \frac{1}{d}x \\ \mathbb{E}[\|\Gamma(x)\|^2] &= d^2\mathbb{E}[\langle x, \mathbf{e}_i \mathbf{e}_i^\top \mathbf{e}_i \mathbf{e}_i^\top x \rangle] = d\|x\|^2.\end{aligned}$$

Next, choose operator \mathcal{B} to be constant; in particular $\mathcal{B}x = \beta x$. Thus to satisfy (13) it suffices to have

$$2\alpha dm + \beta^2 \left(1 - \frac{1}{d}\right) \leq \beta^2(1 - \alpha\sigma) \quad \Rightarrow \quad 2\alpha dm + \alpha\sigma\beta^2 \leq \frac{\beta^2}{d}.$$

To satisfy (14), it suffices to have

$$2\alpha dm + \frac{\beta^2}{d} \leq 1.$$

It remains to notice that $\beta^2 = \frac{d}{2}$ and $\alpha = \frac{1}{4md+\sigma d}$ satisfies the above conditions.

O.2.2 Setup for Corollary H.2

Note that

$$\mathbb{E}[\mathcal{S}(x)] = \mathbf{D}(\mathbf{p})x$$

and

$$\mathbb{E}[\|\Gamma(x)\|^2] = \|x\|_{\mathbb{E}[\sum_{i \in L} \frac{1}{\mathbf{p}_i} \mathbf{e}_i \mathbf{e}_i^\top \sum_{i \in L} \frac{1}{\mathbf{p}_i} \mathbf{e}_i \mathbf{e}_i^\top]}^2 = \|x\|_{\mathbf{p}}^2.$$

Let us consider \mathcal{B} to be the operator corresponding to left multiplication with matrix $\mathbf{D}(b)$: $\mathcal{B}(x) = \mathbf{D}(b)x$. Thus, for (13) it suffices to have for all i

$$2\alpha m_i \mathbf{p}_i^{-1} + b_i^2(1 - \mathbf{p}_i) \leq b_i^2(1 - \alpha\sigma) \quad \Rightarrow \quad 2\alpha m_i \mathbf{p}_i^{-1} + b_i^2\alpha\sigma \leq b_i^2 \mathbf{p}_i$$

For (14) it suffices to have for all i

$$2\alpha m_i \mathbf{p}_i^{-1} + b_i^2 \mathbf{p}_i \leq 1$$

It remains to notice that choice $b_i^2 = \frac{1}{2\mathbf{p}_i}$ and $\alpha = \min_i \frac{\mathbf{p}_i}{4m_i + \sigma}$ is valid.

O.2.3 Setup for Corollary H.4

Note that

$$\mathbb{E}[\mathcal{S}(x)] = \rho x$$

and

$$\mathbb{E}[\|\Gamma(x)\|^2] = \|x\|_{\mathbb{E}[\sum_{i \in L} \frac{1}{\mathbf{p}_i} \mathbf{e}_i \mathbf{e}_i^\top \sum_{i \in L} \frac{1}{\mathbf{p}_i} \mathbf{e}_i \mathbf{e}_i^\top]}^2 = \|x\|_{\mathbf{p}}^2.$$

Let us consider \mathcal{B} to be the operator corresponding to scalar multiplication with β . Thus, for (54) it suffices to have for all i

$$2\alpha w_i \mathbf{p}_i^{-1} + \beta^2(1 - \rho) \leq \beta^2(1 - \alpha\sigma) \quad \Rightarrow \quad 2\alpha w_i \mathbf{p}_i^{-1} + \beta^2\alpha\sigma \leq \beta^2 \rho.$$

For (55) it suffices to have for all i

$$2\alpha w_i \mathbf{p}_i^{-1} + \beta^2 \rho \leq 1.$$

It remains to notice that choice $\beta^2 = \frac{1}{2\rho}$ and $\alpha = \min_i \frac{1}{4w_i \mathbf{p}_i^{-1} + \sigma \rho^{-1}}$ is valid.

O.3 Setup for Corollary I.1

Choose \mathcal{B} to be operator which maps everything into 0. On top of that, by construction we have $\mathcal{R} = 0$ and thus (13) is satisfied for free. Moreover, from (32) we have (following the steps from Section O.1.2):

$$\mathbb{E} \left[\|\Gamma(\mathcal{M}^{\frac{1}{2}}(\mathbf{X}))\|^2 \right] \leq \sum_{j=1}^n p_j^{-1} v_j \|\mathbf{X}_{:j}\|^2.$$

Further, due to (33) and (34), to satisfy (14) we shall have

$$\frac{2\alpha}{n^2} \sum_{j=1}^n p_j^{-1} v_j \|\mathbf{Y}_{:j}\|^2 \leq \frac{1}{n} \|\mathbf{Y}\|^2$$

which simplifies to

$$\frac{2\alpha}{n} \frac{v_j}{p_j} \leq 1$$

and thus it suffices to choose $\alpha = \frac{n}{2} \min_j \frac{p_j}{v_j}$.

Remark O.1. Factor 2 can be omitted since for Lemma E.6, the second factor is 0 and thus we no longer need the Jensen's inequality.

O.4 Setup for Corollary J.1

First note that $\mathbb{E}[\mathcal{S}(\mathbf{X})] = \rho \mathbf{X}$.

Next, due to (34), (33) with choice $\mathbf{Y} = \mathcal{M}^{\frac{1}{2}} \mathbf{X}$ (13) and (14) become respectively:

$$\frac{2\alpha}{n^2} \mathbb{E} \left[\left\| \sum_{j \in R} p_j^{-1} \mathbf{M}_j^{\frac{1}{2}} \mathbf{Y}_{:j} \right\|^2 \right] + \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}(\mathbf{Y}) \right\|^2 \leq (1 - \alpha\sigma) \|\mathcal{B}(\mathbf{Y})\|^2 \quad (38)$$

$$\frac{2\alpha}{n^2} \mathbb{E} \left[\left\| \sum_{j \in R} p_j^{-1} \mathbf{M}_i^{\frac{1}{2}} \mathbf{Y}_{:i} \right\|^2 \right] + \left\| (\mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}(\mathbf{Y}) \right\|^2 \leq \frac{1}{n} \|\mathbf{Y}\|^2 \quad (39)$$

Note next that

$$\mathbb{E} \left[\left\| \sum_{j \in R} p_j^{-1} \mathbf{M}_j^{\frac{1}{2}} \mathbf{Y}_{:j} \right\|^2 \right] = \mathbb{E} \left[\left\| \sum_{j \in R} \mathbf{M}_j^{\frac{1}{2}} (p_j^{-1} \mathbf{Y}_{:j}) \right\|^2 \right] \leq \sum_{j=1}^n p_j^{-1} v_j \|\mathbf{Y}_{:j}\|^2$$

where we used ESO assumption (32) in the last bound above.

Next choose \mathcal{B} to be multiplication with scalar β . Thus, for (38) it suffices to have for all $j \in [n]$

$$\frac{2\alpha}{n^2} v_j p_j^{-1} + \beta^2 (1 - \rho) \leq \beta^2 (1 - \alpha\sigma) \quad \Rightarrow \quad \frac{2\alpha}{n^2} v_j p_j^{-1} + \beta^2 \alpha\sigma \leq \beta^2 \rho$$

For (39) it suffices to have for all $j \in [n]$

$$\frac{2\alpha}{n^2} v_j p_j^{-1} + \beta^2 \rho \leq \frac{1}{n}$$

It remains to notice that choice $\beta^2 = \frac{1}{2n\rho}$ and $\alpha = \min_j \frac{n}{4v_j p_j^{-1} + n\sigma\rho^{-1}}$ is valid.

O.5 Methods with Bernoulli \mathcal{U} : Proofs

O.5.1 Setup for Corollary K.1

Note first that the choice of \mathcal{S}, \mathcal{U} yield

$$\begin{aligned}\mathbb{E}[\mathcal{S}(x)] &= \rho x \\ \mathbb{E}[\|\Gamma(x)\|^2] &= \mathbb{E}[\|\mathcal{U}x\|^2] = \delta^{-1}\|x\|^2\end{aligned}$$

Next, consider \mathcal{B} operator to be a multiplication with a constant b .

Thus for (13) we should have

$$2\alpha\delta^{-1}L + b^2(1 - \rho) \leq (1 - \alpha\sigma)b^2$$

and for (14) we should have

$$2\alpha\delta^{-1}L + \rho b^2 \leq 1$$

It remains to notice that choices $\alpha = \frac{1}{4\delta^{-1}L + \sigma\rho^{-1}}$ and $b^2 = \frac{1}{2\rho}$ are valid to satisfy the above bounds.

O.5.2 Setup for Corollary K.3

Note first that the choice of \mathcal{S}, \mathcal{U} yields

$$\begin{aligned}\mathbb{E}[\mathcal{S}(\mathbf{X})] &= \mathbf{X}\mathbf{D}(p) \\ \mathbb{E}[\|\Gamma(\mathbf{X})\|^2] &= \mathbb{E}[\|\mathcal{U}(\mathbf{X})e\|^2] = \delta^{-1}\|\mathbf{X}e\|^2 \leq \delta^{-1}n\|\mathbf{X}\|^2\end{aligned}$$

Next, as we have no prior knowledge about $\mathbf{G}(x^*)$, consider \mathcal{R} to be identity operator; i.e. $\text{Range}(\mathcal{R}) = \mathbb{R}^{d \times n}$. Lastly, consider \mathcal{B} operator to be a right multiplication with $\mathbf{D}(b)$.

Thus for (13) we should have

$$\forall j : \quad \frac{2\alpha}{n}\delta^{-1}m + \alpha\sigma b_j^2 \leq b_j^2 p_j$$

and for (14) we should have

$$\forall j : \quad \frac{2\alpha}{n}\delta^{-1}m + p_j b_j^2 \leq \frac{1}{n}$$

It remains to notice that choices $\alpha = \min_j \frac{1}{4\delta^{-1}m + \sigma p_j^{-1}}$ and $b_j^2 = \frac{1}{2n p_j}$ are valid to satisfy the above bounds.

O.5.3 Setup for Corollary K.4

Note first that the choice of \mathcal{S}, \mathcal{U} yields

$$\begin{aligned}\mathbb{E}[\mathcal{S}(x)] &= p \circ x \\ \mathbb{E}[\|\Gamma(x)\|^2] &= \mathbb{E}[\|\mathcal{U}(x)\|^2] = \delta^{-1}\|x\|^2\end{aligned}$$

Next, as we have no prior knowledge about $\mathbf{G}(x^*)$, consider \mathcal{R} to be identity operator; i.e. $\text{Range}(\mathcal{R}) = \mathbb{R}^{d \times n}$. Lastly, consider \mathcal{B} operator to be left multiplication with matrix $\mathbf{D}(b)$.

Thus for (13) we should have

$$\forall i : \quad 2\alpha\delta^{-1}m + b_i^2\alpha\sigma \leq b_i^2 p_i$$

and for (14) we should have

$$2\alpha\delta^{-1}m + p_i b_i^2 \leq 1$$

It remains to notice that choices $\alpha = \min_i \frac{1}{4\delta^{-1}m + \sigma p_i^{-1}}$ and $b_i^2 = \frac{1}{2p_i^{-1}}$ are valid to satisfy the above bounds.

O.6 Combination of left and right sketches (in different operators): Proofs

O.6.1 Setup for Corollary L.1

Note first that the choice of \mathcal{S}, \mathcal{U} yields

$$\begin{aligned} \mathbb{E}[S(\mathbf{X})] &= \mathbf{X}\mathbf{D}(p), \\ \mathbb{E}[\|\Gamma(\mathbf{X})\|^2] &= \|\mathcal{M}^{\frac{1}{2}}(\mathbf{X})e\|_{\mathbf{D}(p^{-1})}^2 \leq n \sum_{j=1}^n \|\mathbf{M}_j \mathbf{X}_{:j}\|_{\mathbf{D}(p^{-1})}^2 = n \sum_{j=1}^n \|\mathbf{X}_{:j}\|_{\mathbf{D}(m^j p^{-1})}^2. \end{aligned}$$

Let \mathcal{B} be right multiplication by $\mathbf{D}(b)$. Thus for (13) we should have

$$\forall i, j : \quad 2\frac{\alpha}{n}m_i^j p_i^{-1} + b_j^2 \alpha \sigma \leq b_j^2 p_j$$

and for (14) we should have

$$\forall i, j : \quad 2\frac{\alpha}{n}m_i^j p_i^{-1} + p_j b_j^2 \leq \frac{1}{n}.$$

It remains to notice that choices $\alpha = \min_{i,j} \frac{1}{4m_i^j p_i^{-1} + \sigma p_j^{-1}}$ and $b_j^2 = \frac{1}{2p_j n}$ are valid to satisfy the above bounds.

O.6.2 Setup for Corollary L.2

Note first that the choice of \mathcal{S}, \mathcal{U} yields

$$\begin{aligned} \mathbb{E}[S(\mathbf{X})] &= \mathbf{D}(p)\mathbf{X} \\ \mathbb{E}[\|\Gamma(\mathbf{X})\|^2] &\leq \sum_{j=1}^n p_j^{-1} v_j \|\mathbf{X}_{:j}\|^2 \end{aligned}$$

The second inequality is a direct consequence of ESO (which is shown in Section O.1.2).

Let \mathcal{B} be left multiplication by $\mathbf{D}(b)$. Thus for (13) we should have

$$\forall i, j : \quad 2\frac{\alpha}{n}v_j p_j^{-1} + b_i^2 \alpha \sigma \leq b_i^2 p_i$$

and for (14) we should have

$$\forall i, j : \quad 2\frac{\alpha}{n}v_j p_j^{-1} + p_i b_i^2 \leq \frac{1}{n}$$

It remains to notice that choices $\alpha = \min_{i,j} \frac{1}{4v_j p_j^{-1} + \sigma p_i^{-1}}$ and $b_i^2 = \frac{1}{2p_i n}$ are valid to satisfy the above bounds.

O.7 Joint Sketches: Proofs

O.7.1 Setup for Corollary M.1

For notational simplicity, denote $\mathbf{M}'^{\frac{1}{2}} \in \mathbb{R}^{d \times n}$ to be the matrix with j -th column equal to (element-wise) square root of m_j . We have

$$\mathbb{E}[\mathcal{S}(\mathbf{X})] = (\mathbf{p}\mathbf{p}^\top) \circ \mathbf{X}$$

and

$$\begin{aligned} \mathbb{E} \left[\|\Gamma(\mathcal{M}^{\frac{1}{2}} \mathbf{X})\|^2 \right] &= \mathbb{E} \left[\left\| \left((\mathbf{p}^{-1} \mathbf{p}^{-1\top}) \circ \left(\left(\sum_{i \in L} \mathbf{e}_i \mathbf{e}_i^\top \right) (\mathbf{M}'^{\frac{1}{2}} \circ \mathbf{X}) \left(\sum_{j \in R} \mathbf{e}_j \mathbf{e}_j^\top \right) \right) \right) \mathbf{e} \right\|^2 \right] \\ &= \mathbb{E} \left[\left\| \left(\left(\sum_{i \in L, j \in R} \mathbf{e}_i \mathbf{e}_j^\top \right) \circ (\mathbf{p}^{-1} \mathbf{p}^{-1\top}) \circ \mathbf{M}'^{\frac{1}{2}} \circ \mathbf{X} \right) \mathbf{e} \right\|^2 \right] \\ &= \mathbb{E} \left[\left\| \left(\sum_{i \in L} \mathbf{e}_i \mathbf{e}_i^\top \right) \left((\mathbf{p}^{-1} \mathbf{p}^{-1\top}) \circ \mathbf{M}'^{\frac{1}{2}} \circ \mathbf{X} \right) \mathbf{e}_R \right\|^2 \right] \\ &= \mathbb{E}_R \left[\left\| \left((\mathbf{p}^{-\frac{1}{2}} \mathbf{p}^{-1\top}) \circ \mathbf{M}'^{\frac{1}{2}} \circ \mathbf{X} \right) \mathbf{e}_R \right\|^2 \right] \\ &= \mathbb{E}_R \left[\text{Trace} \left(\left((\mathbf{p}^{-\frac{1}{2}} \mathbf{p}^{-1\top}) \circ \mathbf{M}'^{\frac{1}{2}} \circ \mathbf{X} \right) \mathbf{I}_{R,R} \left((\mathbf{p}^{-\frac{1}{2}} \mathbf{p}^{-1\top})^\top \circ \mathbf{M}'^{\frac{1}{2}\top} \circ \mathbf{X}^\top \right) \right) \right] \\ &= \text{Trace} \left(\left((\mathbf{p}^{-\frac{1}{2}} \mathbf{p}^{-1\top}) \circ \mathbf{M}'^{\frac{1}{2}} \circ \mathbf{X} \right) \mathbf{P} \left((\mathbf{p}^{-\frac{1}{2}} \mathbf{p}^{-1\top})^\top \circ \mathbf{M}'^{\frac{1}{2}\top} \circ \mathbf{X}^\top \right) \right) \\ &= \text{Trace} \left(\left((\mathbf{p}^{-\frac{1}{2}} \mathbf{e}^\top) \circ \mathbf{M}'^{\frac{1}{2}} \circ \mathbf{X} \right) \mathbf{D}(\mathbf{p})^{-1} \mathbf{P} \mathbf{D}(\mathbf{p})^{-1} \left((\mathbf{p}^{-\frac{1}{2}} \mathbf{e}^\top)^\top \circ \mathbf{M}'^{\frac{1}{2}\top} \circ \mathbf{X}^\top \right) \right) \\ &\leq \text{Trace} \left(\left((\mathbf{p}^{-\frac{1}{2}} \mathbf{e}^\top) \circ \mathbf{M}'^{\frac{1}{2}} \circ \mathbf{X} \right) \mathbf{D}(\mathbf{q})^{-1} \left((\mathbf{p}^{-\frac{1}{2}} \mathbf{e}^\top)^\top \circ \mathbf{M}'^{\frac{1}{2}\top} \circ \mathbf{X}^\top \right) \right) \\ &= \left\| \mathbf{X} \circ \mathbf{M}'^{\frac{1}{2}} \circ (\mathbf{p}^{-\frac{1}{2}} \mathbf{q}^{-\frac{1}{2}\top}) \right\|^2. \end{aligned} \tag{40}$$

Next, choose operator \mathbf{B} to be such that $\mathcal{B}(\mathbf{X}) := \mathbf{B} \circ \mathbf{X}$ for $\mathbf{B} \in \mathbb{R}^{d \times n}$. Thus, for (13) and (14) we shall have respectively

$$\forall i, j : \quad \frac{2\alpha}{n^2} \left(\frac{m_i^j}{\mathbf{p}_i \mathbf{q}_j} \right) + \mathbf{B}_{i,j}^2 \alpha \sigma \leq \mathbf{B}_{ij}^2 \mathbf{p}_i \mathbf{q}_j$$

and

$$\forall i, j : \quad \frac{2\alpha}{n^2} \left(\frac{m_i^j}{\mathbf{p}_i \mathbf{q}_j} \right) + \mathbf{B}_{ij}^2 \mathbf{p}_i \mathbf{q}_j \leq \frac{1}{n}.$$

It remains to choose $\mathbf{B}_{i,j}^2 = \frac{1}{2n\mathbf{p}_i \mathbf{q}_j}$ and $\alpha = \min_{i,j} \frac{n\mathbf{p}_i \mathbf{q}_j}{4m_i^j + n\sigma}$.

O.7.2 Setup for Corollary M.2

We have

$$\mathbb{E}[\mathcal{S}(\mathbf{X})] = \rho \mathbf{X}.$$

Next, choose operator \mathbf{B} to be such that $\mathcal{B}(\mathbf{X}) := \beta \circ \mathbf{X}$ for scalar β which would be specified soon. Proceeding with bound (40), for (13) and (14) we shall have respectively

$$\forall i, j : \quad \frac{2\alpha}{n^2} \left(\frac{m_i^j}{\textcolor{red}{p}_i \textcolor{blue}{q}_j} \right) + \beta^2 \alpha \sigma \leq \beta^2 \rho$$

and

$$\forall i, j : \quad \frac{2\alpha}{n^2} \left(\frac{m_i^j}{\textcolor{red}{p}_i \textcolor{blue}{q}_j} \right) + \beta^2 \rho \leq \frac{1}{n}.$$

It remains to choose $\beta^2 = \frac{1}{2n\rho}$ and $\alpha = \min_{i,j} \frac{1}{4 \frac{m_i^j}{n \textcolor{red}{p}_i \textcolor{blue}{q}_j} + \rho^{-1} \sigma}$.

O.7.3 Setup for Corollary M.3

For notational simplicity, denote $\mathbf{M}' \in \mathbb{R}^{d \times n}$ to be a matrix with j -th column equal to m_j .

Let $\Gamma_t(\mathbf{X}_{:N_t}) = \left(\textcolor{red}{p}^{t-1} \textcolor{blue}{p}^{t-1 \top} \right) \circ \left(\left(\sum_{i \in L_t} \textcolor{red}{e}_i \textcolor{red}{e}_i^\top \right) \mathbf{X}_{:N_t} \left(\sum_{j \in R_t} \textcolor{blue}{e}_j \textcolor{blue}{e}_j^\top \right) \right) \textcolor{blue}{e}_{N_t}$. Thus

$$\mathbb{E}[\mathcal{S}(\mathbf{X})] = \sum_{t=1}^T \left(\textcolor{red}{p}^t \textcolor{red}{p}^{t \top} \right) \circ \mathbf{X}_{:N_t}$$

and

$$\begin{aligned} \mathbb{E}[\|\Gamma(\mathbf{X})\|^2] &= \mathbb{E} \left[\left\| \sum_{t=1}^T \Gamma_t(\mathbf{X}_{:N_t}) \right\|^2 \right] \\ &= \mathbb{E} \left[\left\| \sum_{t=1}^T \Gamma_t(\mathbf{X}_{:N_t}) - \mathbb{E} \left[\sum_{t=1}^T \Gamma_t(\mathbf{X}_{:N_t}) \right] \right\|^2 \right] + \left\| \mathbb{E} \left[\sum_{t=1}^T \Gamma_t(\mathbf{X}_{:N_t}) \right] \right\|^2 \\ &= \mathbb{E} \left[\left\| \sum_{t=1}^T (\Gamma_t(\mathbf{X}_{:N_t}) - \mathbf{X}_{:N_t} \textcolor{blue}{e}_{N_t}) \right\|^2 \right] + \|\mathbf{X} \textcolor{blue}{e}\|^2 \\ &= \sum_{t=1}^T \mathbb{E} \left[\|\Gamma_t(\mathbf{X}_{:N_t}) - \mathbf{X}_{:N_t} \textcolor{blue}{e}_{N_t}\|^2 \right] + \|\mathbf{X} \textcolor{blue}{e}\|^2 \\ &\leq \sum_{t=1}^T \mathbb{E} \left[\|\Gamma_t(\mathbf{X}_{:N_t})\|^2 \right] + \|\mathbf{X} \textcolor{blue}{e}\|^2 \\ &\leq \sum_{t=1}^T \mathbb{E} \left[\|\Gamma_t(\mathbf{X}_{:N_t})\|^2 \right] + n \|\mathbf{X}\|^2. \end{aligned} \tag{41}$$

Using the bounds from Section O.7.1 we further get

$$\mathbb{E} \left[\|\Gamma(\mathcal{M}^{\frac{1}{2}} \mathbf{X})\|^2 \right] \stackrel{(41)+(40)}{\leq} \sum_{t=1}^T \left\| \mathbf{X}_{:N_t} \circ \left(\textcolor{red}{p}^{t-\frac{1}{2}} \textcolor{blue}{q}^{t-\frac{1}{2} \top} \right) \circ \mathbf{M}'_{:N_t} \right\|^2 + n \|\mathbf{M}' \circ \mathbf{X}\|^2.$$

Next, choose operator \mathbf{B} to be such that for any \mathbf{X} : $\mathcal{B}(\mathbf{X}) := \mathbf{B} \circ \mathbf{X}$ where $\mathcal{B} \in \mathbb{R}^{d \times n}$. Thus, for (13) and (14) we shall have respectively

$$\forall i, t, j \in N_t : \quad \frac{2\alpha}{n^2} m_i^j \left(\frac{1}{\textcolor{red}{p}_i^t \textcolor{blue}{q}_j^t} + n \right) + \mathbf{B}_{i,j}^2 \alpha \sigma \leq \mathbf{B}_{i,j}^2 \textcolor{red}{p}_i^t \textcolor{blue}{q}_j^t$$

and

$$\forall i, t, j \in N_t : \quad \frac{2\alpha}{n^2} m_i^j \left(\frac{1}{\textcolor{red}{p}_i^t \textcolor{blue}{q}_j^t} + n \right) + \mathbf{B}_{i,j}^2 \textcolor{red}{p}_i^t \textcolor{blue}{q}_j^t \leq \frac{1}{n}.$$

It remains to choose $\mathbf{B}_{i,j}^2 = \frac{1}{2n\bar{p}^t q_j^t}$ and $\alpha = \min_{j \in N_{t,i,t}} \frac{1}{4m_i^j \left(1 + \frac{1}{n\bar{p}^t q_j^t}\right) + \frac{\sigma}{\bar{p}^t q_j^t}}$.

O.8 Setup for Corollary N.1

Let x be column-wise vectorization of \mathbf{X} . Note that

$$\Gamma(\mathcal{M}^{\frac{1}{2}}(\mathbf{X})) = \mathcal{M}^{\frac{1}{2}}(\mathbf{X}) \mathbf{R} \mathbb{E}[\mathbf{R}]^{-1} \mathbf{e} = \left(\mathbf{e}^\top \mathbb{E}[\mathbf{R}]^{-1} \mathbf{R} \otimes \mathbf{I}_d \right) \begin{pmatrix} \mathbf{M}_1^{\frac{1}{2}} & & \\ & \ddots & \\ & & \mathbf{M}_n^{\frac{1}{2}} \end{pmatrix} x.$$

Thus

$$\mathbb{E} \left[\left\| \Gamma(\mathcal{M}^{\frac{1}{2}}(\mathbf{X})) \right\|^2 \right] \leq \|\mathbf{X}\|^2 \eta.$$

Let $\mathcal{B}(\mathbf{X}) = \beta \mathbf{X} \mathbf{B}$. Thus, we have

$$\begin{aligned} (1 - \alpha\sigma) \|\mathcal{B}\mathbf{X}\|^2 - \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}\mathbf{Y} \right\|^2 &= \beta^2 \text{Trace}(\mathbf{X} \mathbf{B}^\top (\mathbb{E}[\mathbf{R}] - \alpha\sigma \mathbf{I}) \mathbf{B} \mathbf{X}^\top) \\ &\leq \beta^2 \lambda_{\min}(\mathbf{B}^\top (\mathbb{E}[\mathbf{R}] - \alpha\sigma \mathbf{I}) \mathbf{B}) \|\mathbf{X}\|^2 \\ &\leq \beta^2 (\lambda_{\min}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}] \mathbf{B}) - \alpha\sigma \lambda_{\max}(\mathbf{B}^\top \mathbf{B})) \|\mathbf{X}\|^2. \end{aligned}$$

Further,

$$\left\| (\mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B} \mathcal{M}^{\dagger \frac{1}{2}} \mathbf{X} \right\|^2 = \beta^2 \text{Trace}(\mathbf{X} \mathbf{B}^\top \mathbb{E}[\mathbf{R}] \mathbf{B} \mathbf{X}^\top) \leq \beta^2 \|\mathbf{X}\|^2 \lambda_{\max}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}] \mathbf{B}).$$

Using the derived bounds together with (34), (33), for conditions (13) and (14) it suffices to have:

$$\frac{2\alpha}{n^2} \eta + \beta^2 \alpha \sigma \lambda_{\max}(\mathbf{B}^\top \mathbf{B}) \leq \beta^2 \lambda_{\min}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}] \mathbf{B}), \quad (42)$$

and

$$\frac{2\alpha}{n^2} \eta + \beta^2 \lambda_{\max}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}] \mathbf{B}) \leq \frac{1}{n}. \quad (43)$$

It remains to notice that choices $\beta^2 = \frac{1}{2n\lambda_{\max}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}] \mathbf{B})}$ and

$$\alpha = \frac{\lambda_{\min}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}] \mathbf{B})}{4n^{-1} \eta \lambda_{\max}(\mathbf{B}^\top \mathbb{E}[\mathbf{R}] \mathbf{B}) + \sigma \lambda_{\max}(\mathbf{B}^\top \mathbf{B})}$$

are valid.

P Tighter Rates for Left \mathcal{U} by Exploiting Prox

In order to obtain the tightest possible convergence rate, consider \mathbf{W} to be such projection matrix operator that for all k we have

$$\mathbf{W}(x^k) = x^k \quad \text{and} \quad \mathbf{W}(x^*) = x^*. \quad (44)$$

We would also require the proximal operator to be contractive under \mathbf{W} .

Assumption P.1. Suppose that for all x, y

$$\|\text{prox}_{\alpha\psi}(x) - \text{prox}_{\alpha\psi}(y)\|^2 \leq \|x - y\|_{\mathbf{W}}^2 \quad (45)$$

Remark P.2. If \mathbf{W} is identity, Assumption P.1 holds since ψ is convex. Most of the results of the paper use \mathbf{W} as identity. However, exploiting a structure of ψ to design non-trivial \mathbf{W} can speed up Algorithm 1, especially when it comes to \mathcal{U} being left sketch. Specifically, exploiting \mathbf{W} for a specific instance of SEGA yields SAGA-AS, see Section R.

We state a generalization of Theorem 5.2 which takes into consideration \mathbf{W} .

Theorem P.3 (Extension of Theorem 5.2). Consider a setup from Theorem 5.2 and suppose that Assumption P.1 holds.

Next, let α and \mathcal{B} are such that for every $\mathbf{X} \in \mathbb{R}^{d \times n}$ we have

$$\frac{2\alpha}{n^2} \mathbb{E} [\|\mathcal{U}\mathbf{X}e\|_{\mathbf{W}}^2] + \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}\mathcal{M}^\dagger \mathbf{X} \right\|^2 \leq (1 - \alpha\sigma) \|\mathcal{B}\mathcal{M}^\dagger \mathbf{X}\|^2 \quad (46)$$

and for any $\mathbf{X} \in \text{Range}(\mathcal{R})^\perp$

$$\frac{2\alpha}{n^2} \mathbb{E} [\|\mathcal{U}\mathbf{X}e\|_{\mathbf{W}}^2] + \left\| (\mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}\mathcal{M}^\dagger \mathbf{X} \right\|^2 \leq \frac{1}{n} \|\mathcal{M}^\dagger \mathbf{X}\|^2. \quad (47)$$

Proof: For simplicity of notation, in this proof, all expectations are conditional on x^k , i.e., the expectation is taken with respect to the randomness of g^k .

Since

$$x^* = \text{prox}_{\alpha\psi}(x^* - \alpha\nabla f(x^*)), \quad (48)$$

and since the prox operator is non-expansive, we have

$$\begin{aligned} \mathbb{E} [\|x^{k+1} - x^*\|^2] &\stackrel{(48)}{=} \mathbb{E} [\|\text{prox}_{\alpha\psi}(x^k - \alpha g^k) - \text{prox}_{\alpha\psi}(x^* - \alpha\nabla f(x^*))\|^2] \\ &\stackrel{(45)+(44)}{\leq} \mathbb{E} [\|x^k - x^* - \alpha\mathbf{W}(g^k - \nabla f(x^*))\|^2] \\ &\stackrel{(8)}{=} \|x^k - x^*\|^2 - 2\alpha \langle \nabla f(x^k) - \nabla f(x^*), x^k - x^* \rangle \\ &\quad + \alpha^2 \mathbb{E} [\|g^k - \nabla f(x^*)\|_{\mathbf{W}}^2] \\ &\stackrel{(10)+(15)}{\leq} (1 - \alpha\sigma) \|x^k - x^*\|^2 + \alpha^2 \mathbb{E} [\|g^k - \nabla f(x^*)\|_{\mathbf{W}}^2] \\ &\quad - 2\alpha D_f(x^k, x^*). \end{aligned} \quad (49)$$

Since $f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x)$, in view of (15) and (19) we have

$$\begin{aligned} D_f(x^k, x^*) &\stackrel{(15)}{=} \frac{1}{n} \sum_{i=1}^n D_{f_i}(x^k, x^*) \stackrel{(19)}{\geq} \frac{1}{2n} \sum_{i=1}^n \|\nabla f_i(x^k) - \nabla f_i(x^*)\|_{\mathbf{M}_i^\dagger}^2 \\ &= \frac{1}{2n} \|\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*))\|^2. \end{aligned} \quad (50)$$

By combining (49) and (50), we get

$$\begin{aligned} \mathbb{E} \left[\|x^{k+1} - x^*\|^2 \right] &\leq (1 - \alpha\sigma) \|x^k - x^*\|^2 + \alpha^2 \mathbb{E} \left[\|g^k - \nabla f(x^*)\|_{\mathbf{W}}^2 \right] \\ &\quad - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2. \end{aligned}$$

Next, applying Lemma E.6 with $\mathbf{Q} = \mathbf{W}$ leads to the estimate

$$\begin{aligned} \mathbb{E} \left[\|x^{k+1} - x^*\|^2 \right] &\leq (1 - \alpha\sigma) \|x^k - x^*\|^2 - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2 \\ &\quad + \frac{2\alpha^2}{n^2} \mathbb{E} \left[\left\| \mathcal{U} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) e \right\|_{\mathbf{W}}^2 \right] \\ &\quad + \frac{2\alpha^2}{n^2} \mathbb{E} \left[\left\| \mathcal{U} (\mathbf{J}^k - \mathbf{G}(x^*)) e \right\|_{\mathbf{W}}^2 \right]. \end{aligned} \quad (51)$$

Adding α -multiple of (31) for $\mathcal{C} = \mathcal{M}^{\dagger \frac{1}{2}}$ to (51) yields

$$\begin{aligned} &\mathbb{E} \left[\|x^{k+1} - x^*\|^2 \right] + \alpha \mathbb{E} \left[\left\| \mathcal{B} \left(\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^{k+1} - \mathbf{G}(x^*)) \right) \right\|^2 \right] \\ &\leq (1 - \alpha\sigma) \|x^k - x^*\|^2 + \frac{2\alpha^2}{n^2} \mathbb{E} \left[\left\| \mathcal{U} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) e \right\|_{\mathbf{W}}^2 \right] + \\ &\quad \frac{2\alpha^2}{n^2} \mathbb{E} \left[\left\| \mathcal{U} (\mathbf{J}^k - \mathbf{G}(x^*)) e \right\|_{\mathbf{W}}^2 \right] + \alpha \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \left(\mathcal{B} \left(\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^k - \mathbf{G}(x^*)) \right) \right) \right\|^2 \\ &\quad + \alpha \left\| \mathbb{E}[\mathcal{S}]^{\frac{1}{2}} \left(\mathcal{B} \left(\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right) \right) \right\|^2 - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2 \\ &\stackrel{(46)}{\leq} (1 - \alpha\sigma) \|x^k - x^*\|^2 + (1 - \alpha\sigma) \alpha \mathbb{E} \left[\left\| \mathcal{B} \left(\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^k - \mathbf{G}(x^*)) \right) \right\|^2 \right] \\ &\quad + \frac{2\alpha^2}{n^2} \mathbb{E} \left[\left\| \mathcal{U} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) e \right\|_{\mathbf{W}}^2 \right] + \alpha \left\| \mathbb{E}[\mathcal{S}]^{\frac{1}{2}} \left(\mathcal{B} \left(\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right) \right) \right\|^2 \\ &\quad - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2 \\ &\stackrel{(47)}{\leq} (1 - \alpha\sigma) \left(\|x^k - x^*\|^2 + \alpha \mathbb{E} \left[\left\| \mathcal{B} \left(\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^k - \mathbf{G}(x^*)) \right) \right\|^2 \right] \right). \end{aligned}$$

Above, we have used (46) with $\mathbf{Y} = (\mathbf{J}^k - \mathbf{G}(x^*))$ and (47) with $\mathbf{Y} = (\mathbf{G}(x^k) - \mathbf{G}(x^*))$.

Q Even faster rates for SEGA

In this section we exploit a tighter convergence analysis from Section P for SEGA. In short, Section P assumes existence of projection matrix \mathbf{W} such that $\forall k : \mathbf{W}(x^k) = x^k$, $\mathbf{W}(x^*) = x^*$ and $\|\text{prox}_{\alpha\psi}(x) - \text{prox}_{\alpha\psi}(y)\|^2 \leq \|x - y\|_{\mathbf{W}}^2$, and exploits this correspondingly.

We also extend the analysis of SEGA to block diagonal sketches. In particular, consider partition of $[d]$ into sets N_t for $1 \leq t \leq T$ and smoothness matrix \mathbf{M} which is block diagonal with respect to blocks N_t ¹⁶. Therefore, \mathcal{S} and $\mathcal{M}^{\dagger \frac{1}{2}}$ commute.

In order to provide tight convergence analysis under \mathbf{M} smoothness, we shall consider vector $w = \sum_{i=1}^d e_i w_i$ such that

$$\mathbf{M}^{\frac{1}{2}} \mathbb{E} \left[\sum_{i \in L} \frac{1}{p_i} e_i e_i^\top \mathbf{W} \sum_{i \in L} \frac{1}{p_i} e_i e_i^\top \right] \mathbf{M}^{\frac{1}{2}} \preceq \mathbf{D}(p^{-1} \circ w), \quad (52)$$

and assume that for all k (analogously to (34) and (33))

$$h^k - \nabla f(x^*) \in \text{Range}(\mathbf{M}) \quad \text{and} \quad \nabla f(x^k) - \nabla f(x^*) \in \text{Range}(\mathbf{M}). \quad (53)$$

Corollary Q.1. *Iteration complexity of Algorithm 5 with $\alpha = \min_i \frac{p_i}{4w_i + \sigma}$ is $\max_i \left(\frac{4w_i + \sigma}{p_i \sigma} \right) \log \frac{1}{\epsilon}$.*

Corollary Q.1 indicates up to constant factor optimal choice $w_i \propto p_i$.

Remark Q.2. *Corollary Q.1 provides worse bound than results of standard coordinate descent using ESO inequality. It remains open question whether such results can be obtained for SEGA. We conjecture that the RCD rates with ESO might not be achievable as RCD, unlike SEGA, relies heavily on sparse updates. However, the convergence rate coincides RCD with ESO if \mathbf{M} is diagonal (or block diagonal).*

Q.1 Setup for Corollary Q.1

Note that

$$\mathbb{E}[\mathcal{S}(x)] = \mathbf{D}(p)x$$

Next, (52) implies

$$\mathbb{E} \left[\left\| \Gamma(\mathbf{M}^{\frac{1}{2}} x) \right\|^2 \right] = \|x\|_{\mathbf{M}^{\frac{1}{2}} \mathbb{E} \left[\sum_{i \in L} p_i^{-1} e_i e_i^\top \mathbf{W} \sum_{i \in L} p_i^{-1} e_i e_i^\top \right] \mathbf{M}^{\frac{1}{2}}}^2 \leq \|x\|_{p^{-1} \circ w}^2.$$

Thanks to (53), in order to satisfy (13) and (14) it remains to have (we substituted $y = \mathbf{M}^{\dagger \frac{1}{2}} x$):

$$2\alpha \|y\|_{p^{-1} \circ w}^2 + \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}(y) \right\|^2 \leq (1 - \alpha\sigma) \|\mathcal{B}(y)\|^2 \quad (54)$$

$$2\alpha \|y\|_{p^{-1} \circ w}^2 + \left\| (\mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}(y) \right\|^2 \leq \|y\|^2 \quad (55)$$

Let us consider \mathcal{B} to be the operator corresponding to the left multiplication with matrix $\mathbf{D}(b)$. Thus for satisfy (13) it suffices to have for all $i \in [d]$:

$$2\alpha m_i p_i^{-1} + b_i^2(1 - p_i) \leq b_i^2(1 - \alpha\sigma) \quad \Rightarrow \quad 2\alpha m_i p_i^{-1} + b_i^2 \alpha\sigma \leq b_i^2 p_i$$

For (14) it suffices to have for all $i \in [d]$

$$2\alpha m_i p_i^{-1} + b_i^2 p_i \leq 1$$

It remains to notice that choice $b_i^2 = \frac{1}{2p_i}$ and $\alpha = \min_i \frac{p_i}{4m_i + \sigma}$ is valid.

¹⁶This is a special case of Algorithm 5 when index $i \in N_t$ is sampled if and only if whole block N_t is sampled

R SAGA with Arbitrary Sampling via SEGA

Let $x_1, \dots, x_n \in \mathbb{R}^d$ and define $x' = [x_1^\top, \dots, x_n^\top]^\top$. Let $I_{x_1=\dots=x_n}$ to be indicator function of the set $x_1 = \dots = x_n$. Suppose that $f'(x') = \frac{1}{n} \sum_{j=1}^n f_j(x_j)$, $\psi'(x') = I_{x_1=\dots=x_n}(x)$.

Then, unregularized instance of (1) and problem

$$\min_{x' \in \mathbb{R}^{nd}} f'(x') + \psi'(x'). \quad (56)$$

are equivalent. For convenience, define $S_j := \{(d-1)j+1, (d-1)j+2, \dots, (d-1)(j+1)\}$ and consider arbitrary sampling R over sets S_j (for $1 \leq j \leq n$).

Lemma R.1. *Assume that f is σ strongly convex and for all j , f_j is \mathbf{M}_j -smooth. Then, f' satisfies Assumption 5.1 with $\sigma' = \frac{\sigma}{n}$ and $\mathbf{M} = \frac{1}{n} \mathbf{D}(\mathbf{M}_j)$. Let $\mathbf{W} := \frac{1}{n} e e^\top \otimes \mathbf{I}_d$. For SEGA-AS applied on (56) with sampling R , condition (44) and Assumption (P.1) holds. Further, running SEGA with R -sampling (Algorithm 5) on (56) is equivalent to running SAGA (Algorithm 3) on f and conditions (32) and (52) are identical with $w = vn^{-2}$. Lastly, conditions (34)+(33) and (53) are identical as well.*

As a direct consequence of Lemma R.1, we obtain the next corollary.

Corollary R.2. *Corollary G.3 for SAGA applied on the unregularized instance of (1) is identical to Corollary Q.1 for SEGA applied on (56) with R -sampling.*

Remark R.3. *Similarly, it can be shown that LSVRG is a special instance of SVRCD, and the analogy of Corollary H.4 via Theorem P.3 (analogy of Corollary H.2 for SEGA) recovers Corollary J.1.*

R.1 Proof of Lemma R.1

Clearly, $\text{prox}_{\alpha\psi} x' = \mathbf{W} x'$ thus $\mathbf{W}(x'^*) = x'^*$, for all k : $\mathbf{W} x'^k = x'^k$ and Assumption (P.1) holds with equality. Next for all $x', y' \in \text{Range}(\mathbf{W})$ we have

$$\begin{aligned} f'(x') = f'(\mathbf{W}(x')) &= \frac{1}{n} \sum_{j=1}^n f_j(x) \geq \frac{1}{n} \sum_{j=1}^n f_j(y) + \left\langle \nabla \left(\frac{1}{n} \sum_{j=1}^n f_j(y) \right), x - y \right\rangle + \frac{\sigma}{2} \|x - y\|^2 \\ &= f(y') + \langle \nabla f(y'), x' - y' \rangle + \frac{\sigma}{2n} \|x' - y'\|^2. \end{aligned}$$

Similarly

$$\begin{aligned} f'(x') = \frac{1}{n} \sum_{j=1}^n f_j(x) &\leq \frac{1}{n} \sum_{j=1}^n f_j(y) + \left\langle \nabla \left(\frac{1}{n} \sum_{j=1}^n f_j(y) \right), x - y \right\rangle + \sum_{j=1}^n \frac{1}{2n} \|x - y\|_{\mathbf{M}_j}^2 \\ &= f(y') + \langle \nabla f(y'), x' - y' \rangle + \frac{1}{2n} \|x' - y'\|_{\mathbf{M}}^2. \end{aligned}$$

It remains to notice the equivalence of SEGA and SAGA, which follows directly as updates on h^k are identical to updates on $\nabla f_j(\phi_j^k)$ from SAGA.

To show the equivalence of (32) and (52), we shall start with LHS of (52):

$$h'^\top \mathbf{M}^{\frac{1}{2}} \mathbb{E} \left[\sum_{i \in R} \mathbf{p}_i^{-1} \mathbf{e}_i \mathbf{e}_i^\top \mathbf{W} \sum_{i \in R} \mathbf{p}_i^{-1} \mathbf{e}_i \mathbf{e}_i^\top \right] \mathbf{M}^{\frac{1}{2}} h' = \frac{1}{n^2} \mathbb{E} \left[\left\| \sum_{i \in R} \mathbf{M}_i^{\frac{1}{2}} \mathbf{p}_i^{-1} h'_i \right\|^2 \right]$$

which is exactly LHS of (32) with $h_i = \mathbf{p}_i^{-1} h'_i$; and corresponding RHSs are matching as well. Thus we indeed have $w = vn^{-2}$. Lastly, (34)+(33) and (53) are clearly identical in this case.

S Convergence Under Strong Growth Condition

In this section, we extend the result of Algorithm 1 to the case when $F := f + \psi$ satisfies a strong growth condition instead of quasi strong convexity. Note that strong growth is weaker (more general) than quasi strong convexity [14].

Suppose that \mathcal{X}^* is a set of minimizers of convex function F . Clearly, \mathcal{X}^* must be convex. Define $[x]^*$ to be a projection of x onto \mathcal{X}^* .

Assumption S.1. Suppose that F satisfies strong growth, i.e. for every x :

$$F(x) - F([x]^*) \geq \frac{\sigma}{2} \|x - [x]^*\|^2. \quad (57)$$

S.1 Technical proposition and lemma

In order to establish the convergence results, it will be useful to establish Proposition S.2 and Lemma S.3.

Proposition S.2. [37, 27] Let f be \mathbf{M} -smooth and suppose that (57) holds. Suppose that $x^{k+1} = x^k - \alpha g^k$ where $\mathbb{E}[g^k] = \nabla f(x^k)$ and $\alpha \leq \frac{1}{3\lambda_{\max}(\mathbf{M})}$. Then

$$\mathbb{E}_k \left[\|x^{k+1} - [x^{k+1}]^*\|^2 \right] \leq \frac{1}{1 + \sigma\alpha} \mathbb{E}_k \left[\|x^k - [x^k]^*\|^2 \right] + \frac{2\alpha^2}{1 + \sigma\alpha} \mathbb{E}_k \left[\|g^k - \nabla f(x^k)\|^2 \right].$$

Lemma S.3. For any $x^* \in \mathcal{X}^*$ we have

$$\mathbb{E} [\|g^k - \nabla f(x^k)\|^2] \leq \frac{2}{n^2} \mathbb{E} [\|\mathcal{U}(\mathbf{G}(x^*) - \mathbf{J}^k)e\|^2] + \frac{2}{n^2} \mathbb{E} [\|\mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*))e\|^2]. \quad (58)$$

Proof:

$$\begin{aligned} & \mathbb{E} [\|g^k - \nabla f(x^k)\|^2] \\ &= \mathbb{E} \left[\left\| \frac{1}{n} \mathbf{J}^k e - \frac{1}{n} \mathcal{U}(\mathbf{G}(x^k) - \mathbf{J}^k)e - \frac{1}{n} \mathbf{G}(x^k)e \right\|^2 \right] \\ &= \frac{1}{n^2} \mathbb{E} \left[\left\| (\mathbf{J}^k - \mathbf{G}(x^*))e - \mathcal{U}(\mathbf{G}(x^*) - \mathbf{J}^k)e + \mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*))e + (\mathbf{G}(x^*) - \mathbf{G}(x^k))e \right\|^2 \right] \\ &\leq \frac{2}{n^2} \mathbb{E} \left[\left\| (\mathbf{J}^k - \mathbf{G}(x^*))e - \mathcal{U}(\mathbf{G}(x^*) - \mathbf{J}^k)e \right\|^2 \right] \\ &\quad + \frac{2}{n^2} \mathbb{E} \left[\left\| \mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*))e + (\mathbf{G}(x^*) - \mathbf{G}(x^k))e \right\|^2 \right] \\ &\leq \frac{2}{n^2} \mathbb{E} \left[\left\| \mathcal{U}(\mathbf{G}(x^*) - \mathbf{J}^k)e \right\|^2 \right] + \frac{2}{n^2} \mathbb{E} \left[\left\| \mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*))e \right\|^2 \right]. \end{aligned}$$

Lastly, it is necessary to assume the null space consistency of solution set \mathcal{X}^* under \mathbf{M} smothness. A similar assumption was considered in [27].

Assumption S.4. For any $x^*, y^* \in \mathcal{X}$ we have

$$\mathcal{M}^{\dagger \frac{1}{2}} \mathbf{G}(x^*) = \mathcal{M}^{\dagger \frac{1}{2}} \mathbf{G}(y^*). \quad (59)$$

S.2 Theorem

We next state the convergence result of Algorithm 1 under strong growth condition.

Theorem S.5. Suppose that (57) holds. Let \mathcal{B} be any linear operator commuting with \mathcal{S} , and assume $\mathcal{M}^{\dagger^{1/2}}$ commutes with \mathcal{S} . Let \mathcal{R} be any linear operator for which $\mathcal{R}(\mathbf{J}^k) = \mathcal{R}(\mathbf{G}(x^*))$ for every

$k \geq 0$. Define the Lyapunov function Ψ^k as per (12) for any $x^* \in \mathcal{X}^*$. Suppose that $\alpha \leq \frac{1}{\lambda_{\max}(\mathbf{M})}$ and \mathcal{B} are chosen so that

$$\begin{aligned} \frac{2\alpha}{n^2} \left(\frac{3 + \sigma\alpha}{1 + \sigma\alpha} \right) \mathbb{E} \left[\|\mathcal{U}\mathbf{X}e\|^2 \right] + \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} \mathbf{X} \right\|^2 \\ \leq \left(1 - \frac{\alpha\sigma}{2 + 2\alpha\sigma} \right) \left\| \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} \mathbf{X} \right\|^2 \end{aligned} \quad (60)$$

whenever $\mathbf{X} \in \text{Range}(\mathcal{R})^\perp$ and

$$\frac{2\alpha}{n^2} \left(\frac{3 + \sigma\alpha}{1 + \sigma\alpha} \right) \mathbb{E} \left[\|\mathcal{U}\mathbf{X}e\|^2 \right] + \left\| (\mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} \mathbf{X} \right\|^2 \leq \frac{1}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} \mathbf{X} \right\|^2 \quad (61)$$

for all $\mathbf{X} \in \mathbb{R}^{d \times n}$. Then for all $k \geq 0$, we have

$$\mathbb{E} [\Psi^k] \leq \left(1 - \frac{\alpha\sigma}{2 + 2\alpha\sigma} \right)^k \Psi^0.$$

Proof:

Consider any $x^* \in \mathcal{X}^*$. Due to non-expansiveness of the prox operator we have

$$\begin{aligned} \mathbb{E} \left[\|x^{k+1} - [x^{k+1}]^*\|_2^2 \right] &\leq \mathbb{E} \left[\|x^{k+1} - [x^k]^*\|_2^2 \right] \\ &\stackrel{(26)}{=} \mathbb{E} \left[\left\| \text{prox}_{\alpha\psi}(x^k - \alpha g^k) - \text{prox}_{\alpha\psi}([x^k]^* - \alpha \nabla f([x^k]^*)) \right\|_2^2 \right] \\ &\leq \mathbb{E} \left[\|x^k - \alpha g^k - ([x^k]^* - \alpha \nabla f([x^k]^*))\|_2^2 \right] \\ &= \|x^k - [x^k]^*\|_2^2 - 2\alpha \langle \nabla f(x^k) - \nabla f([x^k]^*), x^k - [x^k]^* \rangle \\ &\quad + \alpha^2 \mathbb{E} \left[\|g^k - \nabla f([x^k]^*)\|_2^2 \right] \\ &\stackrel{(17)}{\leq} \|x^k - [x^k]^*\|_2^2 - \frac{2\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}([x^k]^*)) \right\|^2 \\ &\quad + \alpha^2 \mathbb{E}_{\mathcal{D}} \left[\|g^k - \nabla f([x^k]^*)\|_2^2 \right]. \end{aligned}$$

Combining the above bound with Proposition S.2 yields

$$\begin{aligned}
& \mathbb{E} \left[\|x^{k+1} - [x^{k+1}]^*\|_2^2 \right] \\
& \leq \left(\frac{1}{2+2\alpha\sigma} + \frac{1}{2} \right) \|x^k - [x^k]^*\|^2 - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}([x^k]^*)) \right\|^2 \\
& \quad + \frac{1}{2} \alpha^2 \mathbb{E} \left[\|g^k - \nabla f([x^k]^*)\|^2 \right] + \frac{\alpha^2}{1+\sigma\alpha} \mathbb{E} \left[\|g^k - \nabla f(x^k)\|^2 \right] \\
& \leq \left(\frac{\alpha\sigma+2}{2+2\alpha\sigma} \right) \|x^k - [x^k]^*\|^2 - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}([x^k]^*)) \right\|^2 \\
& \quad + \frac{1}{2} \alpha^2 \mathbb{E} \left[\|g^k - \nabla f([x^k]^*)\|^2 \right] + \frac{\alpha^2}{1+\sigma\alpha} \mathbb{E} \left[\|g^k - \nabla f(x^k)\|^2 \right] \\
& \stackrel{(58)}{\leq} \left(\frac{\alpha\sigma+2}{2+2\alpha\sigma} \right) \|x^k - [x^k]^*\|^2 - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}([x^k]^*)) \right\|^2 \\
& \quad + \frac{2\alpha^2}{n^2(1+\sigma\alpha)} \left(\mathbb{E} \left[\|\mathcal{U}(\mathbf{G}(x^*) - \mathbf{J}^k)e\|^2 \right] + \mathbb{E} \left[\|\mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*))e\|^2 \right] \right) \\
& \quad + \frac{1}{2} \alpha^2 \mathbb{E} \left[\|g^k - \nabla f([x^k]^*)\|^2 \right] \\
& \stackrel{(24)}{\leq} \left(\frac{\alpha\sigma+2}{2+2\alpha\sigma} \right) \|x^k - [x^k]^*\|^2 - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}([x^k]^*)) \right\|^2 \\
& \quad + \frac{\alpha^2}{n^2} \left(\frac{2}{1+\sigma\alpha} + 1 \right) \left(\mathbb{E} \left[\|\mathcal{U}(\mathbf{G}(x^*) - \mathbf{J}^k)e\|^2 \right] + \mathbb{E} \left[\|\mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*))e\|^2 \right] \right) \\
& \stackrel{(59)}{\leq} \left(\frac{\alpha\sigma+2}{2+2\alpha\sigma} \right) \|x^k - [x^k]^*\|^2 - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2 \\
& \quad + \frac{\alpha^2}{n^2} \left(\frac{2}{1+\sigma\alpha} + 1 \right) \left(\mathbb{E} \left[\|\mathcal{U}(\mathbf{G}(x^*) - \mathbf{J}^k)e\|^2 \right] + \mathbb{E} \left[\|\mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*))e\|^2 \right] \right).
\end{aligned}$$

Since, by assumption, both \mathcal{B} and $\mathcal{M}^{\dagger \frac{1}{2}}$ commute with \mathcal{S} , so does their composition $\mathcal{A} := \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}}$. Applying Lemma E.5, we get

$$\begin{aligned}
\mathbb{E} \left[\left\| \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^{k+1} - \mathbf{G}(x^*)) \right\|^2 \right] &= \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^k - \mathbf{G}(x^*)) \right\|^2 \\
&\quad + \left\| \mathbb{E}[\mathcal{S}]^{\frac{1}{2}} \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2. \quad (62)
\end{aligned}$$

Adding α multiple of (62) to the previous bounds yields

$$\begin{aligned}
& \mathbb{E} \left[\|x^{k+1} - [x^{k+1}]^*\|_2^2 \right] + \alpha \mathbb{E} \left[\left\| \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^{k+1} - \mathbf{G}(x^*)) \right\|^2 \right] \\
& \leq \left(1 - \frac{\alpha\sigma}{2+2\alpha\sigma} \right) \|x^k - [x^k]^*\|^2 - \frac{\alpha}{n} \left\| \mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2 \\
& \quad + \frac{\alpha^2}{n^2} \left(\frac{3+\sigma\alpha}{1+\sigma\alpha} \right) \left(\mathbb{E} \left[\|\mathcal{U}(\mathbf{G}(x^*) - \mathbf{J}^k)e\|^2 \right] + \mathbb{E} \left[\|\mathcal{U}(\mathbf{G}(x^k) - \mathbf{G}(x^*))e\|^2 \right] \right) \\
& \quad + \alpha \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^k - \mathbf{G}(x^*)) \right\|^2 + \alpha \left\| \mathbb{E}[\mathcal{S}]^{\frac{1}{2}} \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{G}(x^k) - \mathbf{G}(x^*)) \right\|^2 \\
& \stackrel{(61)}{\leq} \left(1 - \frac{\alpha\sigma}{2+2\alpha\sigma} \right) \|x^k - [x^k]^*\|^2 + \frac{\alpha^2}{n^2} \left(\frac{3+\sigma\alpha}{1+\sigma\alpha} \right) \mathbb{E} \left[\|\mathcal{U}(\mathbf{G}(x^*) - \mathbf{J}^k)e\|^2 \right] \\
& \quad + \alpha \left\| (\mathcal{I} - \mathbb{E}[\mathcal{S}])^{\frac{1}{2}} \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^k - \mathbf{G}(x^*)) \right\|^2 \\
& \stackrel{(60)}{\leq} \left(1 - \frac{\alpha\sigma}{2+2\alpha\sigma} \right) \left(\|x^k - [x^k]^*\|^2 + \alpha \left\| \mathcal{B}\mathcal{M}^{\dagger \frac{1}{2}} (\mathbf{J}^k - \mathbf{G}(x^*)) \right\|^2 \right).
\end{aligned}$$

Remark S.6. Since $2 + 2\alpha\sigma = \mathcal{O}(1)$ and $\frac{3\sigma\alpha}{1+\sigma\alpha} = \mathcal{O}(1)$ the convergence rate under strong growth provided by Theorem S.5 is of the same order as the convergence rate under quasi strong convexity (Theorem 5.2).