A Short and Unified Convergence Analysis of the SAG, SAGA, and IAG Algorithms

Feng Zhu, Robert W. Heath Jr., and Aritra Mitra *

Abstract

Stochastic variance-reduced algorithms such as Stochastic Average Gradient (SAG) and SAGA, and their deterministic counterparts like the Incremental Aggregated Gradient (IAG) method, have been extensively studied in large-scale machine learning. Despite their popularity, existing analyses for these algorithms are disparate, relying on different proof techniques tailored to each method. Furthermore, the original proof of SAG is known to be notoriously involved, requiring computer-aided analysis. Focusing on finite-sum optimization with smooth and strongly convex objective functions, our main contribution is to develop a single **unified** convergence analysis that applies to all three algorithms: SAG, SAGA, and IAG. Our analysis features two key steps: (i) establishing a bound on delays due to stochastic sub-sampling using simple concentration tools, and (ii) carefully designing a novel Lyapunov function that accounts for such delays. The resulting proof is short and modular, providing the first high-probability bounds for SAG and SAGA that can be seamlessly extended to non-convex objectives and Markov sampling. As an immediate byproduct of our new analysis technique, we obtain the best known rates for the IAG algorithm, significantly improving upon prior bounds.

1 Introduction

We consider the following finite-sum optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{N}\sum_{i=1}^{N} f_i(x), \tag{1}$$

where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is a continuously differentiable function that is assumed to be L-smooth, f is assumed to be μ-strongly convex, and $x^* = \text{argmin}_{x\in\mathbb{R}^d} f(x)$ is the minimizer of the composite function $f(x)$. Problems of the form in (1) arise in the context of empirical risk minimization in machine learning, where $f(x)$ serves as a finite-sample approximation (based on N samples) of a true risk function (Shalev-Shwartz & Ben-David, 2014).

A natural way to solve (1) is to run the gradient descent (GD) algorithm. When f is L-smooth and μ-strongly convex, GD guarantees exponentially fast convergence to x^*, where the exponent of convergence depends on the *condition number* $\kappa = L/\mu$ (Bubeck et al., 2015). This fast *linear* convergence rate, however, comes at the expense of N gradient evaluations per iteration, which can be computationally demanding when N is large. An appealing alternative is the stochastic gradient descent (SGD) algorithm (Robbins & Monro, 1951) that, at each iteration, moves along

*F. Zhu and A. Mitra are with the Dept. of Electrical and Computer Engineering, North Carolina State University. Email: {fzhu5, amitra2}@ncsu.edu. Robert W. Heath Jr. is with the Dept. of Electrical and Computer Engineering at the University of California, San Diego, USA. Email: rwheathjr@ucsd.edu. This material is based upon work supported in part by the National Science Foundation under Grant No. NSF-CCF-2225555.

the negative gradient of just one component function chosen uniformly at random from the set $[N] := \{1, 2, \ldots, N\}$. While SGD evaluates only one gradient per iteration, the high variance in its update direction necessitates a diminishing step-size sequence to ensure exact convergence to x^* with no residual bias. Unfortunately, this leads to a much slower sublinear rate (Moulines & Bach, 2011).

Variance-Reduction Algorithms. A breakthrough in this regard was achieved by Roux et al. (2012), who invented the stochastic average gradient (SAG) algorithm. Like SGD, SAG evaluates only a single gradient in each iteration, but exploits memory of past gradients of all components to maintain an accurate estimate of the gradient of the composite function f. Remarkably, this approach is able to retain the linear convergence rate of GD. However, the proof of this result in Schmidt et al. (2017) is notoriously challenging, and requires computer-aided analysis. Although a related algorithm called SAGA with a simpler proof was developed by Defazio et al. (2014a), the Lyapunov function in this paper fails to explain the convergence behavior of SAG. A deterministic variant of SAG, called the incremental aggregated gradient (IAG) algorithm, was developed by Blatt et al. (2007), and an explicit linear convergence rate for this algorithm was obtained in Gurbuzbalaban et al. (2017). However, the analysis of IAG, which is fundamentally different from those of SAG and SAGA, yields a much slower rate compared to SAG, SAGA, and GD.

In this context, our main contribution is to develop a **single unified proof** that is surprisingly short and simple and yields linear convergence rates for SAG, SAGA, *and* IAG. Furthermore, our analysis significantly improves the best known rate for IAG. In what follows, we elaborate on our main **contributions**, and discuss their implications in relation to prior work.

1. **Unified Proof Technique.** Stochastic variance-reduced methods like SAG and SAGA, and their deterministic counterparts like IAG, all use memory of past gradients to maintain accurate estimates of the full gradient. However, despite the similarity in their update rules, existing convergence analyses of these algorithms differ considerably. In particular, the difficulty in analyzing SAG has often been attributed to the fact that its update direction is *biased*, unlike those for SGD and SAGA that admit relatively simpler proofs. In Gurbuzbalaban et al. (2017), the authors mention: "*We also note that most of the proofs and proof techniques used in the stochastic setting such as the fact that the expected gradient error is zero do not apply to the deterministic setting and this requires a new approach for analyzing IAG.*" In light of this statement and the preceding discussion, whether a unified analysis framework can explain the dynamics of both biased and unbiased, stochastic and deterministic, variance-reduced (VR) algorithms is far from obvious. We show for the first time that this is indeed possible.

   Our proof framework is simple, intuitive, and modular, and features two key steps. In the first step, for stochastic sub-sampling patterns, we use a concentration argument to control the maximum delay in seeing any component function. As a result, on a "good" event of sufficient measure, one can view both SAG and SAGA as delayed versions of GD, with an upper-bound on the delays that scales as $\tilde{\mathcal{O}}(N)$. Based on this insight, in the second step, we construct a novel Lyapunov function that maintains a window of stale gradients. The careful construction of this function is crucial to us achieving our desired rates. We then establish a one-step contractive recursion for this Lyapunov function, which translates to *high-probability* linear convergence rates for SAG and SAGA in Theorem 3.11. Our Lyapunov function and overall proof structure (outlined above) depart fundamentally from prior analyses of SAG and SAGA.

2. **High-Probability Bounds for SAG and SAGA.** The traditional analyses of stochastic optimization algorithms typically provide bounds that hold only in expectation. This is true for VR algorithms like SAG and SAGA as well, and the proofs of these algorithms in Defazio

et al. (2014a); Roux et al. (2012); Schmidt et al. (2017) only provide in-expectation guarantees. Unfortunately, such guarantees only capture the "typical" behavior of the algorithm, and do not adequately represent rare/tail events. As a result, a series of high-probability bounds have emerged for SGD and its variants under different noise models over the last few years; see Li & Orabona (2020); Liu et al. (2023); Sadiev et al. (2023) for more on this topic. Despite this rich literature, to our knowledge, high-probability bounds for SAG and SAGA have remained elusive. Theorem 3.11 closes this gap and contributes to a deeper understanding of these celebrated VR algorithms.

3. **Bounds under Markov Sampling.** SAG and SAGA have been typically analyzed under I.I.D. (independent and identically distributed) sampling of component functions. Thanks to our modular proof framework, we show in Theorem 3.14 that our analysis extends seamlessly to more general Markov sampling schemes. For arriving at this result, we leverage a Bernstein concentration bound for uniformly ergodic Markov chains from Paulin (2015).

4. **Improved Bounds for IAG.** The work of Blatt et al. (2007) that originally developed the IAG algorithm provided no explicit convergence rates for general smooth and strongly convex functions. This gap was later addressed by Gurbuzbalaban et al. (2017), who, for deterministic cyclic sampling patterns, established a convergence rate of $\mathcal{O}\left(\exp(-K/(\kappa^2 N^2))\right)$ after K iterations of IAG. Here, recall that κ is the condition number and N is the number of component functions. Due to the quadratic dependence on κ and N in the exponent, the rate predicted for IAG in Gurbuzbalaban et al. (2017) is much slower compared to that for GD, SAG, and SAGA. Such a pessimistic picture for IAG has, in fact, prompted the development of more complex deterministic variance-reduction algorithms; see Mokhtari et al. (2018). As a byproduct of our analysis for SAG and SAGA, we establish a significantly tighter rate of $\mathcal{O}\left(\exp(-K/(\kappa N))\right)$ for IAG in Theorem 4.1. Intuitively, this rate makes sense, since N iterations of IAG are comparable to one iteration of GD. Thus, our unified analysis provides a more accurate understanding of the true dynamics of IAG.

As a minor contribution, our results can be extended straightforwardly to the smooth, non-convex setting, as demonstrated in Section 3.4; see Theorem 3.12. The concentration bounds we derive in this regard complement the known in-expectation bounds of VR methods for non-convex losses derived in Reddi et al. (2016a,b). Overall, we anticipate that the simple modular nature of our proof framework can be built upon to develop tail bounds for more complex (e.g., accelerated, second-order) VR algorithms in the future.

More Related Work. The literature on stochastic VR algorithms is vast, and we refer the reader to the excellent survey by Gower et al. (2020). Aside from SAG and SAGA, other popular VR algorithms that enjoy linear convergence rates for smooth, strongly convex functions include SVRG (Johnson & Zhang, 2013), SDCA (Shalev-Shwartz & Zhang, 2013), S2GD (Konečnỳ & Richtárik, 2017), MISO Mairal (2015), FINITO (Defazio et al., 2014b), and SARAH (Nguyen et al., 2017). Notably, different from the high-probability bounds we provide, these papers derive guarantees that hold in expectation. As such, our proof techniques differ from those in these works. For deterministic sampling, incremental gradient (IG) methods (Bertsekas et al., 2011) use only one component function to update the parameter in each iteration; like SGD, they suffer from slow sub-linear rates. Deterministic VR methods like IAG (Blatt et al., 2007) and DIAG (Mokhtari et al., 2018) use memory to achieve linear rates like GD. As far as we are aware, no prior work has unified the analysis of stochastic and deterministic VR methods within a single framework.

2 Technical Background

In this section, we introduce the two celebrated stochastic variance-reduced algorithms that we wish to study: SAG and SAGA. To that end, consider first-order iterative algorithms of the general form below for solving Problem (1):

$$x_{k+1} = x_k - \alpha g_k^{\mathcal{A}}, \qquad 0 \leq k \leq K-1 \tag{2}$$

where $\alpha \in (0,1)$ is the step-size, $x_k \in \mathbb{R}^d$ denotes the parameter at iteration k, $g_k^{\mathcal{A}}$ is an estimate of the full gradient $\nabla f(x_k)$ at iteration k, $\mathcal{A}$ refers to the algorithm under study (e.g., GD, SGD, SAG, SAGA, etc.), and $K > 0$ denotes the number of iterations of the algorithm. For simplicity, we will set $x_0 = 0$. We now discuss some relevant instances of (2), and their performance guarantees when each f_i in (1) is L-smooth, and f is μ-strongly convex. Unless stated otherwise, this will be our running assumption on the functions that appear in (1).

• **Gradient Descent (GD)**. In the GD algorithm, $g_k^{\mathbf{GD}}$ takes the following form:

$$g_k^{\mathbf{GD}} := \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x_k), \tag{3}$$

which is essentially the global (full) gradient $\nabla f(x_k)$. By selecting $\alpha = 1/L$, the convergence rate for GD after K iterations is given by (Bubeck et al., 2015):

$$f(x_K) - f(x^*) \leq \left(1 - \frac{1}{\kappa}\right)^K \left(f(x_0) - f(x^*)\right), \tag{4}$$

where $\kappa := L/\mu$ is the condition number. Although GD enjoys exact convergence to the optimum x^* at a *linear* rate, it suffers from a significant computational bottleneck since N gradients need to be computed at each iteration, where N could be prohibitively large in practice.

• **Stochastic Gradient Descent (SGD)**. A well-studied alternative to GD is SGD, which updates the parameter x_k using a randomly selected component gradient, i.e.,

$$g_k^{\mathbf{SGD}} := \nabla f_{i_k}(x_k), \tag{5}$$

where i_k is sampled in an **I.I.D.** manner from $[N]$ *uniformly at random*. While SGD substantially reduces the per-iteration computational cost, the high variance of the update direction prevents convergence to the exact minimizer x^* under a constant step-size. Specifically, the in-expectation error of SGD after K iterations with step-size $\alpha = \mu/(2L^2)$ is given by (Wright & Recht, 2022)

$$\mathbb{E}\left[\|x_K - x^*\|_2^2\right] \leq \left(1 - \frac{1}{2\kappa^2}\right)^K \|x_0 - x^*\|_2^2 + \frac{1}{N}\sum_{i=1}^{N} \|x_i^* - x^*\|_2^2, \tag{6}$$

where $x_i^* \in \operatorname{argmin}_{x\in\mathbb{R}^d} f_i(x)$, and the term $(1/N)\sum_{i=1}^N \|x_i^* - x^*\|_2^2$ captures a bias effect that arises from sub-sampling. With SGD, even if the iterate sequence is initialized at the point x^*, the algorithm can still keep making updates, since x^* need not be a fixed point for any of the component functions. To ensure convergence to x^*, a diminishing step-size sequence is then needed, which leads to a slower *sub-linear* rate of $\mathcal{O}(1/K)$ (Bubeck et al., 2015; Moulines & Bach, 2011).

• **SAG and SAGA**. To reduce the variance of SGD, the SAG algorithm of Roux et al. (2012) maintains a memory of previously computed component gradients. At each iteration k, SAG selects an index i_k uniformly at random from $[N]$ as in SGD, computes the corresponding component

gradient $\nabla f_{i_k}(x_k)$, and updates the iterate x_k as per (2) using the average of all stored gradients, i.e., with the following update direction:

$$g_k^{\mathbf{SAG}} := \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x_{\tau_{i,k}}) + \frac{\nabla f_{i_k}(x_k) - \nabla f_{i_k}(x_{\tau_{i_k,k}})}{N}. \tag{7}$$

Here, $\tau_{i,k} < k$ (initialized to $\tau_{i,k} = 0$ for all i, k) denotes the most recent iteration *before* iteration k at which the component gradient ∇f_i was evaluated. If ∇f_i has not been accessed prior to iteration k, then $\nabla f_i(x_{\tau_{i,k}})$ is set to zero.

While SAG also computes only one component gradient per iteration as SGD, it manages to achieve linear convergence to x^* using extra storage, with a rate given by:

$$\mathbb{E}\left[f(x_K)\right] - f(x^*) \leq \left(1 - \min\left\{\frac{1}{16\kappa}, \frac{1}{8N}\right\}\right)^K C_0, \tag{8}$$

where C_0 is a constant depending on initialization (Schmidt et al., 2017). While this was a remarkable result at the time, the convergence proof of SAG in Schmidt et al. (2017) is extremely complicated, and requires computer-aided tools to verify the non-negativity of certain polynomials that arise in their potential function. The difficulty in the analysis has often been attributed to the fact that for SAG, $g_k^{\mathbf{SAG}}$ is not *unbiased*, and so the usual descent argument where the expected gradient error is zero does not apply.

To address this, the SAGA algorithm (Defazio et al., 2014a) preserves the sampling and memory mechanism of SAG, but proposes a bias-correction technique that yields an unbiased gradient estimator $g_k^{\mathbf{SAGA}}$, defined as follows:

$$g_k^{\mathbf{SAGA}} := \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x_{\tau_{i,k}}) + \nabla f_{i_k}(x_k) - \nabla f_{i_k}(x_{\tau_{i_k,k}}). \tag{9}$$

The parameter x_k is then updated similarly following (2). The only difference of (9) from (7) is that the correction term $\nabla f_{i_k}(x_k) - \nabla f_{i_k}(x_{\tau_{i_k,k}})$ is added without the $1/N$ scaling factor, successfully removing the bias of the gradient estimator. To see why, let $\mathcal{F}_k$ be the σ-algebra generated by $\{i_0, \ldots, i_k\}$, and observe that

$$\begin{aligned}
&\mathbb{E}\left[g_k^{\mathbf{SAGA}} \mid \mathcal{F}_{k-1}\right] \\
&= \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x_{\tau_{i,k}}) + \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x_k) - \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x_{\tau_{i,k}}) \\
&= \nabla f(x_k),
\end{aligned} \tag{10}$$

where the first equality holds due to uniform sampling. This corroborates the unbiasedness of $g_k^{\mathbf{SAGA}}$. Despite this difference from SAG, SAGA achieves a similar rate as SAG (Defazio et al., 2014a):

$$\mathbb{E}\left[f(x_K)\right] - f(x^*) \leq \left(1 - \frac{1}{2(N+\kappa)}\right)^K C_1, \tag{11}$$

where C_1 is some initialization-dependent constant. While the exponent of convergence of SAGA is similar to that of SAG, and the two algorithms share the same computational and memory costs, the unbiasedness of SAGA leads to a much simpler convergence analysis relative to that for SAG.

Paper Outline. Although the update rules for SAG and SAGA in (7) and (9), respectively, are very similar, and they achieve essentially the same rates, their current analyses are dramatically different. In this context, perhaps surprisingly, we show that a unified proof can be developed to achieve high-probability bounds for both SAG and SAGA. This is the subject of Section 3. We also show that the simplicity of our proof lends itself to more general settings: we consider non-convex objectives in Section 3.4 and Markov sampling in Section 3.5. Finally, in Section 4, we discuss how the proof technique developed in Section 3 for stochastic VR algorithms can be applied, *with no modifications at all*, to deterministic counterparts such as IAG. In the process, we significantly improve prior rates for IAG. Overall, our work sheds novel insights into the dynamics of celebrated variance-reduced algorithms that constitute the workhorse of modern machine learning problems.

3 Analysis and Results

In this section, we develop high-probability bounds for SAG and SAGA for smooth, strongly convex and non-convex objectives. Our proof has two key steps. In the first step (Section 3.1), we use Bernstein's inequality to bound a "gradient-staleness" effect that arises from sub-sampling. Informed by this bound, we construct a novel Lyapunov function and analyze its behavior in the second step (Section 3.3). To keep the paper self-contained, we recall the basic definitions and implications of smoothness and strong-convexity used in the main text. For proofs of the implications, we refer the reader to Bubeck et al. (2015).

Definition 3.1 (Smoothness)**.** A function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if for any $x, y \in \mathbb{R}^d$, the following holds:

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L \|x - y\|_2 , \tag{12}$$

where $\nabla(\cdot)$ denotes the gradient operator.

An immediate consequence of smoothness is the following:

$$f(y) - f(x) \leq \langle y - x, \nabla f(x)\rangle + \frac{L}{2}\|y - x\|_2^2, \forall x, y \in \mathbb{R}^d. \tag{13}$$

Definition 3.2 (Strong Convexity)**.** A function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if the following holds for any $x, y \in \mathbb{R}^d$:

$$f(y) \geq f(x) + \langle \nabla f(x),\, y - x\rangle + \frac{\mu}{2}\|x - y\|_2^2 . \tag{14}$$

The following gradient-domination property is a consequence of strong-convexity:

$$\|\nabla f(y)\|_2^2 \geq 2\mu(f(y) - f(x^*)), \forall y \in \mathbb{R}^d, \text{ where } x^* = \operatorname*{argmin}_{x\in\mathbb{R}^d} f(x). \tag{15}$$

3.1 Step 1: Bounding the Staleness from Sub-Sampling

We start by noting that the main distinction between the GD gradient $g_k^{\mathbf{GD}}$ in (3) and the SAG/SAGA gradients $g_k^{\mathbf{SAG}}, g_k^{\mathbf{SAGA}}$ in (7) and (9) lies in the staleness of the component gradients in the latter, as captured by $\tau_{i,k}$. Our simple yet key observation is that while the staleness in seeing any component i, namely $k - \tau_{i,k}$, is a random object, it is not completely uncontrolled. In particular, using concentration, one can show that *the staleness of component gradients can be upper bounded with high probability*. This observation is formalized in the following lemma.

Lemma 3.3 (Bounded Staleness). *For any $\delta \in (0,1)$ and $\tau \geq (8N/3)\log(NK/\delta)$, with probability at least $1-\delta$, the following holds:*

$$k - \tau_{i,k} \leq \tau, \qquad \forall i \in [N],\ 0 \leq k \leq K-1. \tag{16}$$

Proof. To prove Lemma 3.3, we start by showing that for any fixed component $i \in [N]$ and iteration $k = k_0$, the event that i is not sampled within a window of τ consecutive iterations starting from k_0 occurs with probability at most $\delta/(NK)$. To prove this, we will appeal to Bernstein's inequality, which we record below for completeness (Boucheron et al., 2003).

Bernstein's Inequality. *Let $X_1, \cdots, X_k$ be independent zero-mean random variables (RVs). Suppose that $|X_i| \leq M, \forall i \in [k]$. Then, for all $t > 0$, we have*

$$\mathbb{P}\left(\sum_{i=1}^{k} X_i \leq -t\right) \leq \exp\left(-\frac{\frac{1}{2}t^2}{\sum_{i=1}^{k}\mathbb{E}\left[X_i^2\right] + \frac{1}{3}Mt}\right). \tag{17}$$

With this tool at hand, let us fix a component $i \in [N]$, an iteration $k = k_0$, and define a random variable $Y_{i,k} := \mathbb{I}\{i_k = i\} \in \{0,1\}$, where we use $\mathbb{I}\{\mathcal{H}\}$ to represent the indicator RV for an event $\mathcal{H}$. Fix an integer $\tau > 0$ and note that *within any window of τ iterations* starting from $k = k_0$, the probability that component i is never sampled is given by

$$\mathbb{P}\left(\sum_{k=k_0}^{\tau+k_0-1} Y_{i,k} \leq 0\right).$$

The next thing to do is control this probability using Bernstein's inequality. Under the I.I.D. sampling model, we have $\mathbb{E}\left[Y_{i,k}\right] = p := 1/N$. Defining $X_{i,k} := Y_{i,k} - p$, let us create a sequence $\{X_{i,k}\}$ of zero-mean, bounded, and independent RVs with $\mathbb{E}[X_{i,k}] = 0$ and $|X_{i,k}| \leq M := 1$, for all $i \in [N]$ and $k \geq 0$. Furthermore, let us note that $\mathbb{E}[X_{i,k}^2] = \mathbb{V}[Y_{i,k}] = p(1-p) \leq p$. Using (17), we then have

$$\begin{aligned}
\mathbb{P}\left(\sum_{k=k_0}^{\tau+k_0-1} Y_{i,k} \leq 0\right) &\overset{(a)}{=} \mathbb{P}\left(\sum_{k=k_0}^{\tau+k_0-1} X_{i,k} \leq -\tau p\right) \\
&\overset{(b)}{\leq} \exp\left(-\frac{\frac{1}{2}\tau^2 p^2}{\tau p + \frac{1}{3}\tau p}\right) \\
&\overset{(c)}{=} \exp\left(-\frac{3\tau}{8N}\right),
\end{aligned} \tag{18}$$

where in (a), we use the definition of $X_{i,k}$; in (b), we use (17) and the fact that $M = 1$, $\mathbb{E}[X_{i,k}^2] \leq p$; and in (c), we use $p = 1/N$. Requiring the right-hand-side (RHS) of (18) to be smaller than a prescribed failure probability $\delta/(NK)$, and union-bounding over all N components and K iterations yields the desired claim of the lemma. □

Informed by Lemma 3.3, we set $\tau = \lceil (8N/3)\log(NK/\delta) \rceil$, and define a "good event" $\mathcal{G}$ as follows:

$$\boxed{\mathcal{G} := \{k - \tau_{i,k} \leq \tau,\ \forall i \in [N],\ \forall k \geq 0\}.} \tag{19}$$

From Lemma 3.3, we know that $\mathcal{G}$ has measure at least $1-\delta$. Moreover, on event $\mathcal{G}$, the gradient estimators for SAG and SAGA use information at most τ time-steps old, allowing us to treat SAG and SAGA as first-order methods perturbed by a *uniformly bounded delay sequence*. We will build on this insight for our subsequent analysis, where we will condition on the event $\mathcal{G}$.

3.2 Bounding the Gradient Error

First, let us define

$$e_k := g_k - \nabla f(x_k) \tag{20}$$

as the error in the SAG/SAGA gradient estimator relative to the full gradient, where g_k is either $g_k^{\mathbf{SAG}}$ or $g_k^{\mathbf{SAGA}}$ (we drop the superscript on $g_k^{\mathcal{A}}$ when it applies to both SAG and SAGA). Defining

$$r_k := f(x_k) - f(x^*) \tag{21}$$

as the function sub-optimality gap, we have the following approximate descent lemma that captures the one-step descent in the function gap r_k; the result applies to both SAG and SAGA.

Lemma 3.4 (Approximate Descent)**.** *The following holds for both SAG and SAGA by selecting* $\alpha \leq 1/(4L)$*:*

$$r_{k+1} \leq r_k - \frac{\alpha}{4} \|\nabla f(x_k)\|_2^2 + \alpha \|e_k\|_2^2, \qquad \forall k \geq 0. \tag{22}$$

Proof. Using the smoothness of f, we have:

$$\begin{aligned}
r_{k+1} &\overset{(a)}{\leq} r_k + \langle \nabla f(x_k),\, x_{k+1} - x_k \rangle + \frac{L}{2} \|x_{k+1} - x_k\|_2^2 \\
&\overset{(b)}{=} r_k - \alpha \langle \nabla f(x_k),\, g_k \rangle + \frac{L\alpha^2}{2} \|g_k\|_2^2 \\
&\overset{(c)}{=} r_k - \alpha \langle \nabla f(x_k),\, \nabla f(x_k) + e_k \rangle + \frac{L\alpha^2}{2} \|\nabla f(x_k) + e_k\|_2^2 \\
&\overset{(d)}{\leq} r_k - \alpha \|\nabla f(x_k)\|_2^2 - \alpha \langle \nabla f(x_k),\, e_k \rangle + L\alpha^2 \left(\|\nabla f(x_k)\|_2^2 + \|e_k\|_2^2 \right) \\
&\overset{(e)}{\leq} r_k - \left(\frac{\alpha}{2} - \alpha^2 L \right) \|\nabla f(x_k)\|_2^2 + \left(\frac{\alpha}{2} + \alpha^2 L \right) \|e_k\|_2^2 \\
&\overset{(f)}{\leq} r_k - \frac{\alpha}{4} \|\nabla f(x_k)\|_2^2 + \alpha \|e_k\|_2^2 .
\end{aligned} \tag{23}$$

Here, (a) is a consequence of smoothness of f, and follows from (13); for (b), we use the update formula in (2); (c) uses the definition of e_k; (d) uses the elementary inequality: $\|a+b\|_2^2 \leq 2\|a\|_2^2 + 2\|b\|_2^2, \forall a, b \in \mathbb{R}^d$; (e) employs Young's inequality to control the inner product term, and (f) uses $\alpha \leq 1/(4L)$. This completes the proof. □

Lemma 3.4 shows that the one-step descent is perturbed by the gradient error term $\|e_k\|_2^2$. The following lemma is then motivated by this issue, and provides an upper bound on $\|e_k\|_2^2$ that depends on *a window of past gradients*.

Lemma 3.5 (Gradient Error)**.** *On event* $\mathcal{G}$*, the gradient error* e_k *satisfies the following for both SAG and SAGA:*

$$\|e_k\|_2^2 \leq 4\alpha^2 L^2 \tau U_k, \quad \forall k \geq \tau, \quad \text{with } U_k := \sum_{j=1}^{\tau} \|g_{k-j}\|_2^2 .$$

Proof. We first prove the result for SAG. Fix a time-step $k \geq \tau$. On event $\mathcal{G}$, we know that the staleness in seeing any component is at most τ, and hence, every component has been sampled at least once by time-step k. Thus, on $\mathcal{G}$, $\nabla f_i(x_{\tau_{i,k}})$ is a well defined object for $k \geq \tau$. With this in mind, following the definition of e_k and $g_k^{\mathbf{SAG}}$ in (7), we have

$$\begin{aligned}
\|e_k\|_2 &= \frac{1}{N}\left\|\sum_{i\neq i_k}\nabla f_i(x_{\tau_{i,k}}) + \nabla f_{i_k}(x_k) - \sum_{i\in[N]}\nabla f_i(x_k)\right\|_2 \\
&= \frac{1}{N}\left\|\sum_{i\neq i_k}\left(\nabla f_i(x_{\tau_{i,k}}) - \nabla f_i(x_k)\right)\right\|_2 \\
&\overset{(a)}{\leq} \frac{1}{N}\sum_{i\neq i_k}\left\|\nabla f_i(x_{\tau_{i,k}}) - \nabla f_i(x_k)\right\|_2 \\
&\overset{(b)}{\leq} \frac{1}{N}\sum_{i\neq i_k} L\left\|x_{\tau_{i,k}} - x_k\right\|_2 \\
&\overset{(c)}{\leq} \frac{L}{N}\sum_{i\neq i_k}\sum_{j=1}^{\tau}\|x_{k-j+1} - x_{k-j}\|_2 \\
&\overset{(d)}{\leq} L\alpha\sum_{j=1}^{\tau}\left\|g_{k-j}^{\mathbf{SAG}}\right\|_2,
\end{aligned} \tag{24}$$

where (a) is due to the triangle inequality, (b) uses the smoothness of f_i, (c) uses the triangle inequality and the fact that delays are at most τ conditioned on $\mathcal{G}$ (from Lemma 3.3), and (d) follows from (2). Squaring both sides of (24) and using Jensen's inequality yields the desired claim for SAG.

Similarly, for SAGA, we have

$$\begin{aligned}
\|e_k\|_2 &\leq \left\|\frac{1}{N}\sum_{i=1}^{N}\left(\nabla f_i(x_{\tau_{i,k}}) - \nabla f_i(x_k)\right)\right\|_2 + \left\|\nabla f_{i_k}(x_k) - \nabla f_{i_k}(x_{\tau_{i_k,k}})\right\|_2 \\
&\leq 2L\alpha\sum_{j=1}^{\tau}\left\|g_{k-j}^{\mathbf{SAGA}}\right\|_2,
\end{aligned} \tag{25}$$

where we omit the intermediate steps since they follow exactly as in the SAG analysis in (24). □

Note that Lemma 3.5 holds only for $k \geq \tau$ such that the past gradient terms in U_k are well defined. The next corollary is an immediate consequence of Lemma 3.5 that follows from the definition of e_k and Jensen's inequality.

Corollary 3.6 (Gradient Bound)**.** *On event $\mathcal{G}$, the following holds for both SAG and SAGA, for all $k \geq \tau$:*

$$\|g_k\|_2^2 \leq 2\|\nabla f(x_k)\|_2^2 + 8L^2\tau\alpha^2 U_k. \tag{26}$$

This completes the first step of our analysis. We now proceed to the next step that involves designing an appropriate Lyapunov function.

3.3 Step 2: Designing the Lyapunov Function

From Lemma 3.5, observe that the bound on $\|e_k\|_2^2$ depends on a window of past gradients, and hence, cannot be absorbed by a single-step descent argument as in (22). This motivates the choice of a Lyapunov function with a *shifted window* term that tracks the recent history. To that end, we construct the following Lyapunov function V_k for $k \geq \tau$:

$$V_k := f(x_k) - f(x^*) + L\alpha^2 W_k, \text{ with } W_k := \sum_{j=1}^{\tau} (\tau - j + 1) \|g_{k-j}\|_2^2 . \tag{27}$$

The weight $\tau - j + 1$ assigned to the past gradient $\|g_{k-j}\|_2^2$ is motivated by how past gradients accumulate in the one-step descent analysis. Specifically, for a fixed k, from the descent inequality in Lemma 3.4, and the bound on $\|e_k\|_2^2$ from Lemma 3.5, we note that the stale gradient $\|g_{k-j}\|_2^2$ appears in exactly $\tau - j + 1$ future descent inequalities over the window $[k, k + \tau - 1]$. By assigning larger weights to more recent gradients, our choice of the Lyapunov function in (27) carefully accounts for this multiplicity. In a moment, we will see how this choice allows us to establish a one-step contractive recursion for V_k. To proceed, we will need the following facts about the "shifted window" terms U_k and W_k associated with the Lyapunov function.

Fact 3.7. *The following holds for any $k \geq \tau$:*

$$W_{k+1} = W_k - U_k + \tau \|g_k\|_2^2 , \tag{28}$$

Fact 3.8. *The following holds for any $k \geq \tau$:*

$$W_k \leq \tau U_k. \tag{29}$$

The proofs of these facts follow directly from the definitions of U_k and W_k, and are hence omitted. We now have all the pieces needed to establish a one-step recursion for V_k.

Lemma 3.9 (One-Step Recursion)**.** *Suppose f in (1) is μ-strongly convex. Let $\alpha = 1/(16L\tau)$. Then, on event $\mathcal{G}$, the following is true for both SAG and SAGA:*

$$V_{k+1} \leq \left(1 - \frac{\alpha\mu}{4}\right) V_k, \quad \forall k \geq \tau. \tag{30}$$

Proof. Using (27), plugging the bound on $\|e_k\|_2^2$ in Lemma 3.5 into (22), and adding $L\alpha^2 W_{k+1}$ to both sides of (22) yields the following recursion that holds for both SAG and SAGA:

$$\begin{aligned}
V_{k+1} &\leq r_k - \frac{\alpha}{4} \|\nabla f(x_k)\|_2^2 + 4L^2\tau\alpha^3 U_k + L\alpha^2 W_{k+1} \\
&\overset{(a)}{=} r_k - \frac{\alpha}{4} \|\nabla f(x_k)\|_2^2 + 4L^2\tau\alpha^3 U_k + L\alpha^2 \left(W_k - U_k + \tau \|g_k\|_2^2\right) \\
&\overset{(b)}{\leq} r_k - \frac{\alpha}{4} \|\nabla f(x_k)\|_2^2 + 4L^2\tau\alpha^3 U_k + L\alpha^2 \left(W_k - U_k + 2\tau \|\nabla f(x_k)\|_2^2 + 8L^2\tau^2\alpha^2 U_k\right) \\
&= r_k - \left(\frac{\alpha}{4} - 2L\tau\alpha^2\right) \|\nabla f(x_k)\|_2^2 + L\alpha^2 W_k + \left(4L^2\tau\alpha^3 + 8L^3\tau^2\alpha^4 - L\alpha^2\right) U_k \\
&\overset{(c)}{\leq} r_k - \frac{\alpha}{8} \|\nabla f(x_k)\|_2^2 + L\alpha^2 \left(W_k - \frac{1}{2} U_k\right) \\
&\overset{(d)}{\leq} \left(1 - \frac{\alpha\mu}{4}\right) r_k + L\alpha^2 \left(1 - \frac{1}{2\tau}\right) W_k \\
&\overset{(e)}{\leq} \left(1 - \frac{\alpha\mu}{4}\right) r_k + L\alpha^2 \left(1 - \frac{\alpha\mu}{4}\right) W_k = \left(1 - \frac{\alpha\mu}{4}\right) V_k.
\end{aligned} \tag{31}$$

In the above steps, (a) follows from decomposing W_{k+1} using Fact 3.7; (b) uses Corollary 3.6 to bound $\|g_k\|_2^2$; (c) holds by selecting $\alpha = 1/(16L\tau)$ such that $2L\tau\alpha^2 \leq \alpha/8$, $4L^2\tau\alpha^3 \leq L\alpha^2/4$, and $8L^3\tau^2\alpha^4 \leq L\alpha^2/4$; (d) uses Fact 3.8 to bound U_k and the gradient domination property of strong convexity in (15); and (e) holds by selecting $\alpha = 1/(16L\tau) \leq 2/(\mu\tau)$. □

There is an important caveat here after obtaining inequality (30). Since the bound on $\|e_k\|_2^2$ in Lemma 3.5 only holds for $k \geq \tau$, the one-step Lyapunov recursion in Lemma 3.9 therefore only makes sense for $k \geq \tau$. As such, iterating (30) from $k = \tau$ to $k = K - 1$ yields:

$$f(x_K) - f(x^*) \overset{(*)}{\leq} V_K \leq \left(1 - \frac{\alpha\mu}{4}\right)^{K-\tau} V_\tau, \tag{32}$$

where $(*)$ holds from the definition of V_K. The appearance of V_τ reflects a finite burn-in period where the bounded staleness condition has not yet kicked in. To complete the analysis, we need to argue that at the end of this period, V_τ remains bounded. This is the subject of the next result.

Lemma 3.10 (Burn-In Effects). *Define* $B := \max\{\|x^*\|_2, \|x_1^*\|_2, \cdots \|x_N^*\|_2\}$, *where* $x_i^* \in \operatorname{argmin}_{x\in\mathbb{R}^d} f_i(x)$. *With* $\alpha = 1/(16L\tau)$, *the following is true:*

$$V_\tau \leq 3LB^2. \tag{33}$$

Remark. The dependence of B on the minimizers $\{x_i^*\}$ of the component functions can be explained as follows. Recall that $\tau = \tilde{\mathcal{O}}(N)$, i.e., the initial burn-in period is on the order of the number of components N (up to log factors). As such, there are bound to be certain time-steps $k < \tau$, such that, at time k, not every component function has been sampled at least once. Nonetheless, for both SAG and SAGA, updates to the iterates are still made during the initial burn-in period. As a result, the effective function being optimized during this phase can differ from the true one f, and the iterates may get biased toward the local minimizers. A possible remedy is to modify the existing burn-in phase so that no iterate updates are performed during the first τ iterations, and the algorithm only collects component gradients and updates the memory during this phase. Iterate updates begin after τ, once all components have been sampled at least once on the event $\mathcal{G}$. In this case, one can show that $B = \|x^*\|_2$ suffices to capture burn-in effects.

The proof of Lemma 3.10 can be divided into three steps: (i) we first show by induction that during iterations $0 \leq k \leq \tau$, the iterates are uniformly bounded as follows: $\|x_k\|_2 \leq B$; (ii) using this, we show that the gradient norms for $0 \leq k \leq \tau$ are bounded as follows: $\|g_k\|_2 \leq 6LB$; and (iii) finally, we show that V_τ is bounded by $3LB^2$ using (i), (ii), and the definition of V_τ. The details of the proof are provided in Appendix A.

We now resume our main convergence analysis. Plugging the bound for V_τ in Lemma 3.10 into (32), and selecting $\alpha = 1/(16L\tau)$ yields

$$\begin{aligned} f(x_K) - f(x^*) &\leq 3LB^2 \left(1 - \frac{1}{64\tau\kappa}\right)^K \left(1 - \frac{1}{64\tau\kappa}\right)^{-\tau} \\ &\leq 6LB^2 \left(1 - \frac{1}{64\tau\kappa}\right)^K. \end{aligned}$$

In the last step, we used Bernoulli's inequality: $(1+x)^r \geq 1 + rx$, where $r \geq 1$ is a positive integer and $x \geq -1$. Based on our developments in Sections 3.1– 3.3, and the fact that event $\mathcal{G}$ has measure at least $1 - \delta$, we have established the following result.

Theorem 3.11 (SAG/SAGA, Strongly Convex Case)**.** *Suppose that each f_i in (1) is L-smooth, and f is μ-strongly convex. Given any $\delta \in (0,1)$, let $\tau = \lceil (8N/3)\log(NK/\delta) \rceil$, and set $\alpha = 1/(16L\tau)$. Then, with probability at least $1-\delta$, the following holds for both SAG and SAGA when $K > \tau$:*

$$f(x_K) - f(x^*) \leq 6LB^2 \left(1 - \frac{1}{64\tau\kappa}\right)^K, \tag{34}$$

where $\kappa = L/\mu$ and B is as defined in Lemma 3.10.

Main Takeaways. Our result above reveals that with high-probability, SAG and SAGA converge exponentially fast to the optimal point x^*, where the exponent depends on the product of the condition number κ and the staleness factor τ. As far as we are aware, Theorem 3.11 *is the first high-probability bound that applies identically to both SAG and SAGA*. Notably, our analysis that leads to this result is significantly simpler, shorter, and self-contained compared to the highly involved and computer-aided analysis for SAG in Schmidt et al. (2017).

Since $\tau = \tilde{\mathcal{O}}(N)$, the exponent of convergence in (34) is slower by a factor of N relative to the exponent for gradient descent in (4). Intuitively, this makes sense since N iterations of SAG/SAGA lead to the same number of gradient evaluations as in one step of GD. While the rate in (8) is better than that in (34), we conjecture that this difference arises from the fact that the former is an in-expectation guarantee, while the latter is a high-probability bound. In particular, high-probability bounds need to account for tail events where the delay can indeed be on the order $\tilde{\mathcal{O}}(N)$. To provide further insights about our rate, consider the deterministic delayed GD update rule of the form $x_{k+1} = x_k - \alpha\nabla f(x_{k-\tau})$, where $\tau > 0$ is a constant delay. Interestingly, for this update rule, it is shown by Arjevani et al. (2020) that a rate on the order of $\exp(-K/(\tau\kappa))$ is, in fact, *tight*. In other words, the deterioration of the exponent by the delay τ (relative to GD) is unavoidable. Whether the lower bound in Arjevani et al. (2020) applies to our setting remains to be seen.

3.4 Extension to Non-Convex Objectives

We now show that the analysis for the strongly convex case can be easily generalized to account for non-convex objectives. To see this, observe that just on the basis of smoothness of each f_i, one can arrive at inequality (c) in the chain of inequalities in (31). We then have

$$\begin{aligned} V_{k+1} &\leq r_k - \frac{\alpha}{8}\|\nabla f(x_k)\|_2^2 + L\alpha^2\left(W_k - \frac{1}{2}U_k\right) \\ &\leq V_k - \frac{\alpha}{8}\|\nabla f(x_k)\|_2^2, \end{aligned} \tag{35}$$

where the second inequality follows from discarding the negative term $-L\alpha^2 U_k/2$. Rearranging and telescoping (35) from iteration $k = \tau$ to $k = K-1$ yields

$$\frac{1}{K-\tau}\sum_{k=\tau}^{K-1}\|\nabla f(x_k)\|_2^2 \leq \frac{8V_\tau}{(K-\tau)\alpha} \leq \frac{256L\tau V_\tau}{K}, \tag{36}$$

when $K \geq 2\tau$, and $\alpha = 1/(16L\tau)$. Using the bound on V_τ from Lemma 3.10 then immediately yields the following result for smooth, non-convex functions.

Theorem 3.12 (SAG/SAGA, Non-Convex Case)**.** *Suppose that each f_i in (1) is L-smooth. Given any $\delta \in (0,1)$, let τ and α be as in Theorem 3.11. Then, with probability at least $1-\delta$, the following holds for both SAG and SAGA:*

$$\frac{1}{K-\tau}\sum_{k=\tau}^{K-1}\|\nabla f(x_k)\|_2^2 \leq \frac{768L^2B^2\tau}{K}, \quad \forall K \geq 2\tau. \tag{37}$$

Main Takeaway. For smooth, non-convex objectives, it is well known that gradient descent provides a $\mathcal{O}(1/K)$ convergence rate for the object on the LHS of (37) (Bubeck et al., 2015). Interpreting K/τ as the "effective" number of iterations (due to sub-sampling), Theorem 3.12 establishes a similar high-probability rate for SAG and SAGA. It is worth pointing out that if the function f satisfies the Polyak–Łojasiewicz (PL) condition (Karimi et al., 2016) in (15), then one can obtain the same linear convergence rate as in Theorem 3.11 following an identical analysis.

3.5 Extension to Markov Sampling

Thus far, we have worked under the assumption that at each iteration k, a component i_k is sampled in an I.I.D. manner, uniformly at random from $[N]$. In this section, we will significantly relax such an assumption, and demonstrate that our analysis seamlessly extends to a more general Markov sampling scheme. Specifically, we now consider a scenario where the sampling indices $\{i_k\}_{k\geq 0}$ form the trajectory of a time-homogeneous, aperiodic, and irreducible Markov chain $\mathcal{M}$ supported on $[N]$. Let π be the stationary distribution of this ergodic chain, and, for simplicity, assume that $i_0 \sim \pi$, causing the chain to be stationary.[1]

We note that the basic SGD algorithm has been analyzed under Markov sampling in several papers; for instance, see Doan (2022); Duchi et al. (2012); Sun et al. (2018). Like us, these papers also work under the assumption that the data-generating Markov chain is ergodic. However, to our knowledge, there are no known high-probability bounds for variance-reduced algorithms such as SAG and SAGA under Markov sampling. Our goal is to close this gap.

With this goal in mind, let $\pi_{\min} := \min_{i\in[N]} \pi_i > 0$ denote the smallest entry in the stationary distribution π, representing the *minimum visitation probability*. It should be noted that for our subsequent analysis, we do not require π to be a uniform distribution over $[N]$. As such, our analysis can handle the case when the gradient estimators (for both SAG and SAGA) are *biased*. To build some intuition, let us think back to the analysis under I.I.D. sampling. More than the I.I.D. aspect itself, what mattered was the fact that, with high probability, each component function is visited sufficiently often. This, in turn, ensured a bounded staleness effect, which caused the rest of the analysis to go through. Thus, as long as we can argue that under Markov sampling, a similar bounded staleness property is preserved, the remainder of the analysis will be identical to that of the I.I.D. case. We now show that ergodicity buys us exactly this desired property. To that end, we introduce the *mixing time* function of $\mathcal{M}$ following Dorfman & Levy (2022): $d_{mix}(k) := \sup_{i\in[N]} D_{TV}\left(\mathbb{P}(i_k \in \cdot \mid i_0 = i), \pi\right)$, where D_{TV} is the *total variation distance* between probability measures. Next, we define the mixing time of $\mathcal{M}$ as $t_{mix} := \inf\{k \mid d_{mix}(k) \leq 1/4\}$. Using the objects defined above, we can then prove the following key result.

[1]The assumption of stationarity can be avoided at the expense of more algebra that we omit here for clarity of exposition.

Lemma 3.13 (Bounded Staleness under Markov Sampling). *For any $\delta \in (0,1)$ and $\tau \geq (88 t_{mix}/\pi_{\min}) \log(NK/\delta)$, with probability at least $1-\delta$,*

$$k - \tau_{i,k} \leq \tau, \qquad \forall i \in [N],\ 0 \leq k \leq K-1. \tag{38}$$

The proof of Lemma 3.13 is provided in Appendix B; the key technical tool we use to establish this result is a variant of Bernstein's inequality for Markov sampling developed by Paulin (2015). The only distinction between Lemma 3.13 and Lemma 3.3 lies in the dependence of τ on the minimum visitation probability $\pi_{\min}$ and the mixing time t_{mix}. Informed by Lemma 3.13, define the new staleness parameter as $\tau = \lceil (88 t_{mix}/\pi_{\min}) \log(NK/\delta) \rceil$. Now suppose the good event $\mathcal{G}$ in (19) is defined exactly as before with this new choice of τ. Conditioned on this event $\mathcal{G}$, the remainder of the analysis under Markov sampling is identical to that under I.I.D. sampling carried out in Sections 3.2 and 3.3. As a result, we immediately obtain the following theorem.

Theorem 3.14 (SAG/SAGA, Markov Sampling). *Consider the Markov sampling scheme described in Section 3.5. Suppose that each f_i in (1) is L-smooth, and f is μ-strongly convex. Given any $\delta \in (0,1)$, let $\tau = \lceil (88 t_{mix}/\pi_{\min}) \log(NK/\delta) \rceil$, and set $\alpha = 1/(16L\tau)$. Then, with probability at least $1-\delta$, the following holds for both SAG and SAGA when $K > \tau$:*

$$f(x_K) - f(x^*) \leq 6LB^2 \left(1 - \frac{1}{64\tau\kappa}\right)^K. \tag{39}$$

Main Takeaways. The main takeaway is that our result above under Markov sampling matches that for the I.I.D. case, except for the fact that the staleness parameter τ now depends on the mixing time and the minimum visitation probability of the Markov chain. Essentially, up to log factors, one can now interpret K/t_{cov} as the effective number of iterations, where $t_{\text{cov}} := t_{mix}/\pi_{min}$. Since Theorem 3.14 appears to be the *first high-probability bound for SAG and SAGA under Markov sampling*, we cannot comment on the tightness of our bound in terms of its dependence on t_{cov}. That said, the inflation by a factor of t_{cov} is typically seen for stochastic approximation algorithms subject to Markov sampling, when the Markov chain is supported on a finite state-space; for instance, for tabular Q-learning, see Qu & Wierman (2020). The inflation by the mixing time t_{mix} appears more generally for SGD in Nagaraj et al. (2020), for reinforcement learning algorithms like temporal-difference learning in Bhandari et al. (2018); Mitra (2024); Srikant & Ying (2019), and nonlinear stochastic approximation in Chen et al. (2022).

4 Extension to the IAG Method

Although we have considered stochastic sampling schemes thus far, we now show that our analysis framework can easily accommodate deterministic sampling patterns, as well. To that end, we consider the classical **incremental aggregated gradient (IAG)** method, a *deterministic* counterpart of SAG, introduced by Blatt et al. (2007). The gradient estimator $g_k^{\mathbf{IAG}}$ of IAG has the same aggregated form as SAG in (7), and the iterate is also updated via (2). The key difference is that the component functions are sampled one at a time in *any* deterministic order, such that every component is sampled at least once in every τ iterations, i.e., we have

$$k > \tau_{i,k} \geq k - \tau, \quad \forall i \in [N], \forall k \geq \tau, \tag{40}$$

where $\tau \in \mathbb{N}^+$ is some prescribed parameter, and $\tau_{i,k}$ has the same meaning as before. This condition coincides exactly with the high probability event $\mathcal{G}$ in (19), where the maximum delay in sampling any component is τ. As a result, for the IAG algorithm, event $\mathcal{G}$ occurs with probability 1. Consequently, Theorems 3.11 and 3.12 hold for IAG **deterministically** without any modification. We record this observation below for the strongly convex case.

Theorem 4.1 (IAG, Strongly Convex Case)**.** *Suppose that each f_i in (1) is L-smooth, and f is μ-strongly convex. Consider the IAG method with a sampling pattern that satisfies (40). With $\alpha = 1/(16L\tau)$, the following then holds:*

$$f(x_K) - f(x^*) \leq 6LB^2 \left(1 - \frac{1}{64\tau\kappa}\right)^K, \forall K > \tau. \tag{41}$$

Main Takeaways. Comparing Theorem 4.1 with Theorem 3.11, we note that the deterministic guarantee for IAG is *identical* to the high-probability bound we derived earlier for SAG/SAGA. Such a finding appears to be new. Perhaps more interestingly, our developments so far reveal that a single proof technique suffices to provide a unified treatment of both stochastic and deterministic variance-reduced algorithms. In addition to this unification, a key contribution of our work is that it significantly improves upon the best known convergence rate for IAG, as we discuss below.

Tighter bounds for IAG. As far as we are aware, the best known rate for the IAG method for smooth and strongly convex objectives was derived by Gurbuzbalaban et al. (2017), and is as follows:

$$f(x_K) - f(x^*) \leq \frac{L}{2}\left(1 - \frac{c_\tau}{(\kappa+1)^2}\right)^{2K} \|x_0 - x^*\|_2^2,$$

where $c_\tau := 2/(25\tau(2\tau+1))$. From the above display, we note that while the convergence is still exponential, the exponent scales *quadratically* in both the condition number κ, and the delay τ. In sharp contrast, our analysis for IAG in Theorem 4.1 is able to achieve a *linear* dependence in both κ and τ. This is a significant improvement for ill-conditioned problems. Moreover, note that for a simple cyclic sampling pattern, $\tau = N - 1$. Thus, for modern empirical risk minimization problems, where N represents a potentially large number of data samples, our rate marks a considerable improvement over prior work. Notably, such an improvement is a free byproduct of our unified proof strategy, and requires no extra work beyond what we did for SAG/SAGA.

5 Conclusion

We revisited two of the most popular stochastic VR algorithms, namely SAG and SAGA, and introduced a novel *unified* proof framework that yields linear convergence rates for both. Notably, our proof is considerably shorter and simpler compared to the known analysis for SAG, and leads to the first high-probability bounds for both SAG and SAGA. We show that our analysis can be easily extended to non-convex settings and Markov sampling. Finally, we argue that, as an immediate byproduct of our analysis, one can significantly tighten the best known rates for the IAG algorithm. There are various interesting directions that one can pursue as future work. First, as mentioned in the discussion immediately after Theorem 3.11, there is a gap between our high-probability linear convergence rate and the in-expectation rates for SAG and SAGA in prior work. We would like to investigate further whether this gap is an artifact of our analysis or fundamental. In a similar spirit, deriving lower bounds for the Markov sampling case would be interesting. Finally, extending our proof template to more complex VR algorithms is a natural next step.

References

Arjevani, Y., Shamir, O., and Srebro, N. A tight convergence analysis for stochastic gradient descent with delayed updates. In *Algorithmic Learning Theory*, pp. 111–132. PMLR, 2020.

Bertsekas, D. P. et al. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011.

Bhandari, J., Russo, D., and Singal, R. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on Learning Theory*, pp. 1691–1692. PMLR, 2018.

Blatt, D., Hero, A. O., and Gauchman, H. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2007.

Boucheron, S., Lugosi, G., and Bousquet, O. Concentration inequalities. In *Summer school on machine learning*, pp. 208–240. Springer, 2003.

Bubeck, S. et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Chen, Z., Zhang, S., Doan, T. T., Clarke, J.-P., and Maguluri, S. T. Finite-sample analysis of nonlinear stochastic approximation with applications in reinforcement learning. *Automatica*, 146: 110623, 2022.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, volume 27, 2014a.

Defazio, A., Domke, J., et al. Finito: A faster, permutable incremental gradient method for big data problems. In *International Conference on Machine Learning*, pp. 1125–1133. PMLR, 2014b.

Doan, T. T. Finite-time analysis of markov gradient descent. *IEEE Transactions on Automatic Control*, 2022.

Dorfman, R. and Levy, K. Y. Adapting to mixing time in stochastic optimization with markovian data. In *International Conference on Machine Learning*, pp. 5429–5446. PMLR, 2022.

Duchi, J. C., Agarwal, A., Johansson, M., and Jordan, M. I. Ergodic mirror descent. *SIAM Journal on Optimization*, 22(4):1549–1578, 2012.

Gower, R. M., Schmidt, M., Bach, F., and Richtárik, P. Variance-reduced methods for machine learning. *Proceedings of the IEEE*, 108(11):1968–1983, 2020.

Gurbuzbalaban, M., Ozdaglar, A., and Parrilo, P. A. On the convergence rate of incremental aggregated gradient algorithms. *SIAM Journal on Optimization*, 27(2):1035–1048, 2017.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 795–811. Springer, 2016.

Konečnỳ, J. and Richtárik, P. Semi-stochastic gradient descent methods. *Frontiers in applied mathematics and statistics*, 3:9, 2017.

Li, X. and Orabona, F. A high probability analysis of adaptive sgd with momentum. *arXiv preprint arXiv:2007.14294*, 2020.

Liu, Z., Nguyen, T. D., Nguyen, T. H., Ene, A., and Nguyen, H. High probability convergence of stochastic gradient methods. In *International Conference on Machine Learning*, pp. 21884–21914. PMLR, 2023.

Mairal, J. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

Mitra, A. A simple finite-time analysis of td learning with linear function approximation. *IEEE Transactions on Automatic Control*, 2024.

Mokhtari, A., Gurbuzbalaban, M., and Ribeiro, A. Surpassing gradient descent provably: A cyclic incremental method with linear convergence rate. *SIAM Journal on Optimization*, 28(2):1420–1447, 2018.

Moulines, E. and Bach, F. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in neural information processing systems*, 24, 2011.

Nagaraj, D., Wu, X., Bresler, G., Jain, P., and Netrapalli, P. Least squares regression with markovian data: Fundamental limits and algorithms. *Advances in neural information processing systems*, 33: 16666–16676, 2020.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takáč, M. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, pp. 2613–2621. PMLR, 2017.

Paulin, D. Concentration inequalities for markov chains by marton couplings and spectral methods. 2015.

Qu, G. and Wierman, A. Finite-time analysis of asynchronous stochastic approximation and q-learning. In *Conference on Learning Theory*, pp. 3185–3205. PMLR, 2020.

Reddi, S. J., Hefny, A., Sra, S., Poczos, B., and Smola, A. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pp. 314–323. PMLR, 2016a.

Reddi, S. J., Sra, S., Póczos, B., and Smola, A. Fast incremental method for smooth nonconvex optimization. In *2016 IEEE 55th conference on decision and control (CDC)*, pp. 1971–1977. IEEE, 2016b.

Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Roux, N., Schmidt, M., and Bach, F. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Sadiev, A., Danilova, M., Gorbunov, E., Horváth, S., Gidel, G., Dvurechensky, P., Gasnikov, A., and Richtárik, P. High-probability bounds for stochastic optimization and variational inequalities: the case of unbounded variance. In *International Conference on Machine Learning*, pp. 29563–29648. PMLR, 2023.

Schmidt, M., Le Roux, N., and Bach, F. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, 2017.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Shalev-Shwartz, S. and Zhang, T. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.

Srikant, R. and Ying, L. Finite-time error bounds for linear stochastic approximation and TD learning. In *Conference on Learning Theory*, pp. 2803–2830. PMLR, 2019.

Sun, T., Sun, Y., and Yin, W. On markov chain gradient descent. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Wright, S. J. and Recht, B. *Optimization for data analysis*. Cambridge University Press, 2022.

A Proof of Lemma 3.10

As stated immediately after Lemma 3.10, the proof is divided into three steps: (i) proving bounded iterates, (ii) proving bounded gradients, and (iii) proving bounded V_τ. We start with the first step which establishes boundedness of iterates during the interval $0 \le k \le \tau$.

Claim A.1 (Bounded Iterates)**.** *For $0 \le k \le \tau$, we claim that the following holds for both SAG and SAGA, when $\alpha = 1/(16L\tau)$:*

$$\|x_k\|_2 \le B, \tag{42}$$

where B is as defined in Lemma 3.10.

Proof. We prove this claim by induction and first focus on the analysis of SAG. Since we have assumed $x_0 = 0$, the base case holds trivially. Suppose that this claim holds up to iteration $0 \le k \le \tau - 1$. Then for iteration $k+1$, we have

$$\begin{aligned} x_{k+1} &\overset{(a)}{=} x_k - \frac{\alpha}{N}\left(\sum_{i\neq i_k} \nabla f_i(x_{\tau_{i,k}}) + \nabla f_{i_k}(x_k)\right) \\ &\overset{(b)}{=} x_k - \alpha\nabla f(x_k) - \frac{\alpha}{N}\left(\sum_{i\neq i_k} \nabla f_i(x_{\tau_{i,k}}) + \nabla f_{i_k}(x_k) - \sum_{i=1}^{N}\nabla f_i(x_k)\right) \\ &= x_k - \alpha\nabla f(x_k) - \frac{\alpha}{N}\sum_{i\neq i_k}\left(\nabla f_i(x_{\tau_{i,k}}) - \nabla f_i(x_k)\right), \end{aligned} \tag{43}$$

where (a) uses the definition of $g_k^{\mathbf{SAG}}$, and (b) holds by adding and subtracting $\nabla f(x_k)$. Taking the 2-norm on both sides of the above display and using the triangle inequality yields

$$\begin{aligned} \|x_{k+1}\|_2 &\le \|x_k\|_2 + \alpha\|\nabla f(x_k) - \nabla f(x^*)\|_2 + \frac{\alpha}{N}\sum_{i\neq i_k}\left\|\nabla f_i(x_{\tau_{i,k}}) - \nabla f_i(x_k)\right\|_2 \\ &\overset{(a)}{\le} \|x_k\|_2 + \alpha L\|x_k - x^*\|_2 + \frac{\alpha}{N}\sum_{i\in\mathcal{C}_k, i\neq i_k} L\left\|x_{\tau_{i,k}} - x_k\right\|_2 + \frac{\alpha}{N}\sum_{i\in[N]\setminus\mathcal{C}_k, i\neq i_k}\|\nabla f_i(x_k) - \nabla f_i(x_i^*)\|_2 \\ &\overset{(b)}{\le} \|x_k\|_2 + \alpha L\left(\|x_k\|_2 + \|x^*\|_2\right) + \frac{\alpha}{N}\sum_{i\in\mathcal{C}_k, i\neq i_k} L\left(\left\|x_{\tau_{i,k}}\right\|_2 + \|x_k\|_2\right) \\ &\quad + \frac{\alpha}{N}\sum_{i\in[N]\setminus\mathcal{C}_k, i\neq i_k} L\left(\|x_k\|_2 + \|x_i^*\|_2\right) \\ &\overset{(c)}{\le} (1 + L\alpha)\|x_k\|_2 + 3\alpha LB. \end{aligned} \tag{44}$$

In the above steps, (a) uses smoothness, and $\mathcal{C}_k$ denotes the set of component indices that have been sampled at least once up to iteration k. For a component i that is yet to be sampled, by definition, $\nabla f_i(x_{\tau_{i,k}}) = 0 = \nabla f_i(x_i^*)$. We use this in (a). Inequality (b) holds due to smoothness and (c) uses the definition of B and the induction hypothesis up to iteration k.

Iterating (44) for $k+1$ steps from $k = 0$ yields

$$\begin{aligned} \|x_{k+1}\|_2 &\le (1 + L\alpha)^{k+1}\|x_0\|_2 + 3LB\alpha\sum_{j=0}^{k}(1 + L\alpha)^{k-j} \\ &\le 3LB\alpha\cdot\tau(1 + L\alpha)^{\tau} \le 4LB\tau\alpha \le \frac{B}{4} \le B, \end{aligned} \tag{45}$$

where we used the fact that $x_0 = 0$, $k+1 \leq \tau$, $1+x \leq e^x$, and $\alpha = 1/(16L\tau)$. The proof is then complete for the SAG case.

The proof for the case of SAGA carries through similarly as in the proof of Lemma 3.5. Specifically, the update rule of SAGA can be written as

$$x_{k+1} = x_k - \alpha \nabla f(x_k) - \frac{\alpha}{N} \sum_{i=1}^{N} \left(\nabla f_i(x_{\tau_{i,k}}) - \nabla f_i(x_k) \right) - \alpha \left(\nabla f_{i_k}(x_k) - \nabla f_{i_k}(x_{\tau_{i_k,k}}) \right). \tag{46}$$

Taking the 2-norm on both sides and using the triangle inequality yields

$$\begin{aligned} \|x_{k+1}\|_2 &\leq \|x_k\|_2 + \frac{\alpha}{N} \sum_{i=1}^{N} \left\| \nabla f_i(x_{\tau_{i,k}}) - \nabla f_i(x_k) \right\|_2 + \alpha \left\| \nabla f_{i_k}(x_k) - \nabla f_{i_k}(x_{\tau_{i_k,k}}) \right\|_2 \\ &\quad + \alpha \left\| \nabla f(x_k) - \nabla f(x^*) \right\|_2 \\ &\leq (1 + L\alpha) \|x_k\|_2 + 5LB\alpha. \end{aligned} \tag{47}$$

The rationale is similar to the SAG case and is hence omitted, where one needs to break the analysis into two cases: (i) component i has been sampled before iteration k, and (ii) it has never been sampled before iteration k. As has been shown for the SAG analysis, both cases yield the exact same bound. Iterating (47) for $k+1$ steps from $k = 0$ yields the same bound. The proof is then complete. □

The next claim provides an upper bound on the gradients $\|g_k\|_2$ when $0 \leq k \leq \tau$.

Claim A.2 (Bounded Gradients)**.** *For $0 \leq k \leq \tau$, the following holds for both SAG and SAGA with $\alpha = 1/(16L\tau)$:*

$$\|g_k\|_2 \leq 6LB. \tag{48}$$

Proof. For SAG, we can write

$$\begin{aligned} \left\| g_k^{\mathbf{SAG}} \right\|_2 &\leq \frac{1}{N} \sum_{i \neq i_k} \left\| \nabla f_i(x_{\tau_{i,k}}) - \nabla f_i(x_k) \right\|_2 + \left\| \nabla f(x_k) - \nabla f(x^*) \right\|_2 \\ &\leq 2LB + 2LB = 4LB \leq 6LB, \end{aligned} \tag{49}$$

where we used smoothness, Claim A.1, and the same arguments used to bound the gradients in the proof of Claim A.1.

In a similar way, for SAGA, we can write

$$\begin{aligned} \left\| g_k^{\mathbf{SAGA}} \right\|_2 &\leq \frac{1}{N} \sum_{i=1}^{N} \left\| \nabla f_i(x_{\tau_{i,k}}) - \nabla f_i(x_k) \right\|_2 + \left\| \nabla f_{i_k}(x_k) - \nabla f_{i_k}(x_{\tau_{i_k,k}}) \right\|_2 + \left\| \nabla f(x_k) - \nabla f(x^*) \right\|_2 \\ &\leq 2LB + 2LB + 2LB = 6LB. \end{aligned} \tag{50}$$

The claim is then proved. □

With the two claims that we have established in this section, we can bound V_τ as

$$\begin{aligned} V_\tau &= f(x_\tau) - f(x^*) + L\alpha^2 \sum_{j=1}^{\tau} (\tau - j + 1) \left\| g_{\tau-j} \right\|_2^2 \\ &\leq \frac{L}{2} \left\| x_\tau - x^* \right\|_2^2 + L\alpha^2\tau^2 \cdot 36L^2B^2 \\ &\leq \frac{L}{2} \cdot 4B^2 + 36L^3\tau^2B^2\alpha^2 \\ &\leq 2LB^2 + \frac{36}{256} LB^2 \leq 3LB^2, \end{aligned} \tag{51}$$

where we used smoothness, Claim A.1 and A.2.

B Proof of Lemma 3.13

To establish Lemma 3.13, we will leverage a variant of Bernstein's inequality for Markov chains, that we state below (slightly adapted for our purpose).

Theorem B.1. *(Paulin, 2015, Theorem 3.4) [Bernstein's Inequality for Markov Chains] Let $X_1, \cdots, X_k$ be a time-homogeneous, ergodic, and stationary Markov chain $\mathcal{M}$ that takes values in a finite state space Ω. Let γ_{ps} and π be the pseudo spectral gap and stationary distribution, respectively, of $\mathcal{M}$. Suppose f is a measurable function in $L^2(\pi)$, satisfying $|f(x) - \mathbb{E}_\pi[f]| \leq M, \forall x \in \Omega$, where $\mathbb{E}_\pi$ is the expectation w.r.t. π. Then, for all $t > 0$, the following is true:*

$$\mathbb{P}\left(S - \mathbb{E}_\pi[S] \leq -t\right) \leq \exp\left(-\frac{t^2 \cdot \gamma_{ps}}{8\left(k + 1/\gamma_{ps}\right) V_f + 20tM}\right), \tag{52}$$

where $S := \sum_{i=1}^k f(X_i)$, and $V_f := \mathbb{V}_\pi[f]$ is the variance of f under π.

According to Proposition 3.4 of Paulin (2015), the pseudo spectral gap can be related to the mixing time t_{mix} defined in Section 3.5 as:

$$\gamma_{ps} \geq \frac{1}{2t_{mix}}. \tag{53}$$

As a result, if $k \geq t_{mix}$, we then have

$$\begin{aligned} \mathbb{P}\left(S - \mathbb{E}_\pi[S] \leq -t\right) &\leq \exp\left(-\frac{t^2}{16t_{mix}\left(k + 2t_{mix}\right) V_f + 40t_{mix}Mt}\right) \\ &\leq \exp\left(-\frac{t^2}{48kt_{mix}V_f + 40t_{mix}Mt}\right). \end{aligned} \tag{54}$$

Now we are in a position to apply this inequality to our setting where the sampling indices correspond to the X_i's in Theorem B.1. Fix a component $i \in [N]$ and set

$$f(x) = \mathbb{I}\{x = i\}. \tag{55}$$

Like in the proof of Lemma 3.3, we consider the event that component i is not sampled within any window of length τ starting from $k = k_0$. To that end, define S_{i,τ,k_0} as

$$S_{i,\tau,k_0} = \sum_{k=k_0}^{k_0+\tau-1} f(i_k) = \sum_{k=k_0}^{k_0+\tau-1} \mathbb{I}\{i_k = i\}, \tag{56}$$

which counts visits to component i in a length-τ window starting from $k = k_0$. Since $\mathbb{E}_\pi [S_{i,\tau,k_0}] = \tau\pi_i$, where π_i is the entry of component i in π, we have

$$\{S_{i,\tau,k_0} \le 0\} \iff \{S_{i,\tau,k_0} - \mathbb{E}_\pi[S_{i,\tau,k_0}] \le -\tau\pi_i\}. \tag{57}$$

On the RHS of (54), we have $M = 1$ since

$$|f(x) - \mathbb{E}_\pi[f]| = \max\{\pi_i, 1 - \pi_i\} \le 1. \tag{58}$$

We can also compute V_f as

$$V_f = \mathbb{V}_\pi [\mathbb{I}\{x = i\}] = \pi_i(1 - \pi_i) \le \pi_i. \tag{59}$$

With the specifications above, we can apply (54) as follows:

$$\begin{aligned}
\mathbb{P}(S_{i,\tau,k_0} \le 0) &= \mathbb{P}(S_{i,\tau,k_0} - \mathbb{E}_\pi[S_{i,\tau,k_0}] \le -\tau\pi_i) \\
&\le \exp\left(-\frac{\tau^2\pi_i^2}{48\tau t_{mix}\pi_i + 40\tau t_{mix}\pi_i}\right) \\
&= \exp\left(-\frac{\tau\pi_i}{88 t_{mix}}\right) \\
&\le \exp\left(-\frac{\tau\pi_{\min}}{88 t_{mix}}\right),
\end{aligned} \tag{60}$$

where in the last step we used the definition of $\pi_{\min}$. Requiring (60) to be smaller than $\delta/(NK)$, and then union bounding over all components $i \in [N]$ and iterations $0 \le k \le K - 1$ yields the desired claim in Lemma 3.13.