Practical Schemes for Finding Near-Stationary Points of Convex Finite-Sums

Anonymous Author(s)
Affiliation
Address
email

Abstract

The problem of finding near-stationary points in convex optimization has not been adequately studied yet, unlike other optimality measures such as the function value. Even in the deterministic case, the optimal method (OGM-G, due to Kim and Fessler [33]) has just been discovered recently. In this work, we conduct a systematic study of algorithmic techniques for finding near-stationary points of convex finite-sums. Our main contributions are several algorithmic discoveries: (1) we discover a memory-saving variant of OGM-G based on the performance estimation problem approach [19]; (2) we design a new accelerated SVRG variant that can simultaneously achieve fast rates for minimizing both the gradient norm and function value; (3) we propose an adaptively regularized accelerated SVRG variant, which does not require the knowledge of some unknown initial constants and achieves near-optimal complexities. We put an emphasis on the simplicity and practicality of the new schemes, which could facilitate future developments.

1 Introduction

Classic convex optimization usually focuses on providing guarantees for minimizing function value. For this task, the optimal (up to constant factors) Nesterov's accelerated gradient method (NAG) [40, 41] has been known for decades, and there are even methods that can exactly match the lower complexity bounds [30, 17, 55, 18]. On the other hand, in general non-convex optimization, near-stationarity is the typical optimality measure, and there has been a flurry of recent research devoted to this topic [25, 26, 23, 28, 21, 60]. Recently, there has been growing interest on devising fast schemes for finding near-stationary points in convex optimization [42, 2, 22, 7, 31, 32, 33, 27, 15, 14]. This line of research is basically driven by the following facts.

- Nesterov [42] studied the problem with a linear constraint: $f(x^\star) = \min_{x\in Q}\{f(x) : Ax = b\}$, where Q is a convex set and f is strongly convex. Assuming that Q and f are simple, we can focus on the dual problem $\phi(y^\star) = \max_y\{\phi(y) \triangleq \min_{x\in Q}\{f(x) + \langle y, b - Ax\rangle\}\}$. Clearly, the dual objective $-\phi(y)$ is smooth convex. Letting x_y be the unique solution to the inner problem, we have $\nabla\phi(y) = b - Ax_y$. Note that $f(x_y) - f(x^\star) = \phi(y) - \langle y, \nabla\phi(y)\rangle - \phi(y^\star) \leq \|y\|\,\|\nabla\phi(y)\|$. Thus, in this problem, the quantity $\|\nabla\phi(y)\|$ serves as a measure of both primal optimality $f(x_y) - f(x^\star)$ and feasibility $\|b - Ax_y\|$, which is better than just measuring the function value.
- Matrix scaling [50] is a convex problem and its goal is to find near-stationary points [4, 9].
- Gradient norm is readily available, unlike other optimality measures ($f(x) - f(x^\star)$ and $\|x - x^\star\|$), and is thus usable as a stopping criterion. This fact motivates the design of several parameter-free algorithms [43, 39, 27], and their guarantees are established on the gradient norm.
- Designing schemes for minimizing the gradient norm can inspire new non-convex optimization methods. For example, SARAH [46] was designed for convex finite-sums with gradient-norm measure, but was later discovered to be the near-optimal method for non-convex finite-sums [21, 47].

Table 1: Finding near-stationary points $\|\nabla f(x)\| \le \epsilon$ of convex finite-sums.

	Algorithm	Complexity	Remark
IFC	GD [33]	$O(\frac{n}{\epsilon^2})$	
	Regularized NAG* [7]	$O(\frac{n}{\epsilon}\log\frac{1}{\epsilon})$	
	OGM-G [33]	$O(\frac{n}{\epsilon})$	$O(\frac{1}{\epsilon}+d)$ memory, optimal in ϵ
	M-OGM-G [Section 3.1]	$O(\frac{n}{\epsilon})$	$O(d)$ memory, optimal in ϵ
	L2S [37]	$O(n+\frac{\sqrt{n}}{\epsilon^2})$	Loopless variant of SARAH [46]
	Regularized Katyusha* [2]	$O((n+\frac{\sqrt{n}}{\epsilon})\log\frac{1}{\epsilon})$	Requires the knowledge of Δ_0
	R-Acc-SVRG-G* [Section 5]	$O((n\log\frac{1}{\epsilon}+\frac{\sqrt{n}}{\epsilon})\log\frac{1}{\epsilon})$	Without the knowledge of Δ_0
IDC	GD [42, 54]	$O(\frac{n}{\epsilon})$	
	NAG / NAG + GD [32] / [42]	$O(\frac{n}{\epsilon^{2/3}})$	
	Regularized NAG* [42, 27]	$O(\frac{n}{\sqrt{\epsilon}}\log\frac{1}{\epsilon})$	
	NAG + OGM-G [45]	$O(\frac{n}{\sqrt{\epsilon}})$	$O(\frac{1}{\sqrt{\epsilon}}+d)$ memory, optimal in ϵ
	NAG + M-OGM-G [Section 3.1]	$O(\frac{n}{\sqrt{\epsilon}})$	$O(d)$ memory, optimal in ϵ
	Katyusha + L2S [Appendix E]	$O(n\log\frac{1}{\epsilon}+\frac{\sqrt{n}}{\epsilon^{2/3}})$	
	Acc-SVRG-G [Section 4]	$O(n\log\frac{1}{\epsilon}+\frac{n^{2/3}}{\epsilon^{2/3}})$[1]	$O(n\log\frac{1}{\epsilon}+\sqrt{\frac{n}{\epsilon}})$ for function at the same time, simple and elegant
	Regularized Katyusha* [2]	$O((n+\sqrt{\frac{n}{\epsilon}})\log\frac{1}{\epsilon})$	Requires the knowledge of R_0
	R-Acc-SVRG-G* [Section 5]	$O((n\log\frac{1}{\epsilon}+\sqrt{\frac{n}{\epsilon}})\log\frac{1}{\epsilon})$	Without the knowledge of R_0

* Indirect methods (using regularization).

Moreover, finding near-stationary points is a harder task than minimizing function value, because NAG has the optimal guarantee for $f(x)-f(x^\star)$ but is only suboptimal for minimizing $\|\nabla f(x)\|$.

In this work, we consider the problem $\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$, where each f_i is L-smooth and convex. We focus on finding an ϵ-stationary point of this objective, i.e., a point with $\|\nabla f(x)\| \le \epsilon$. We use $\mathcal{X}^\star$ to denote the set of optimal solutions, which is assumed to be nonempty. There are two different assumptions on the initial point x_0, namely, the Initial bounded-Function Condition (**IFC**): $f(x_0) - f(x^\star) \le \Delta_0$, and the Initial bounded-Distance Condition (**IDC**): $\|x_0 - x^\star\| \le R_0$ for some $x^\star \in \mathcal{X}^\star$. This subtlety results in drastically different best achievable rates as studied in [7, 22]. Below we categorize existing algorithmic techniques into three classes (relating to Table 1).

(i) *"IDC + IFC"*. Nesterov [42] showed that we can combine the guarantees of a method minimizing function value under IDC and a method finding near-stationary points under IFC to produce a faster one for minimizing gradient norm under IDC. For example, NAG produces $f(x_{K_1}) - f(x^\star) = O(\frac{LR_0^2}{K_1^2})$ [40] and GD produces $\|\nabla f(x_{K_2})\|^2 = O\big(\frac{L(f(x_0)-f(x^\star))}{K_2}\big)$ [33] under IFC. Letting $x_0 = x_{K_1}$ and $K = K_1 + K_2$, by balancing the ratio of K_1 and K_2, we obtain the guarantee $\|\nabla f(x_K)\|^2 = O(\frac{L^2R_0^2}{K^3})$ for "NAG + GD". We point out that we can use this technique to combine the guarantees of Katyusha [1] and SARAH[2] [46]; see Appendix E.

(ii) *Regularization.* Nesterov [42] used NAG (strongly convex variant) to solve the regularized objective, and showed that it achieves near-optimal complexity (optimal up to logarithmic factors). Inspired by this technique, Allen-Zhu [2] proposed recursive regularization for stochastic approximation algorithms, which also achieves near-optimal complexities [22].

[1] Table 1 shows that Katyusha+L2S has a slightly better dependence on n than Acc-SVRG-G. It is due to the adoption of n-dependent step size in L2S. As studied in [37], despite having a better complexity, n-dependent step size boosts numerical performance only when n is *extremely large*. If the practically fast n-independent step size is used for L2S, Katyusha+L2S and Acc-SVRG-G have the same complexity. See also Appendix A.

[2] We adopt the loopless variant of SARAH in [37], which has a refined analysis for general convex objectives.

(iii) *Direct methods.* Due to the lack of insight, existing direct methods are mostly derived or analyzed with the help of computer-aided tools [31, 32, 54, 33]. The computer-aided approach was pioneered by Drori and Teboulle [19], who introduced the performance estimation problem (PEP). The only known optimal method OGM-G [33] was designed based on the PEP approach.

Observe that since $f(x) - f(x^\star) \leq \|\nabla f(x)\| \|x - x^\star\|$, the lower bound for finding near-stationary points must be of the same order as for minimizing function value [44]. Thus, under IDC, the lower bound is $\Omega(n + \sqrt{\frac{n}{\epsilon}})$ due to [58]. Under IFC, we can establish an $\Omega(n + \frac{\sqrt{n}}{\epsilon})$ lower bound using the techniques in [7, 58]. The main contributions of this work are three new algorithmic schemes that improve the practicalities of existing methods as summarized below (highlighted in Table 1).

- (Section 3) We propose a memory-saving variant of OGM-G for the deterministic case ($n = 1$), which does not require a pre-computed and stored parameter sequence. The derivation of the new variant is inspired by the numerical solution to a PEP problem.
- (Section 4) We propose a new accelerated SVRG [29, 59] variant that can *simultaneously* achieve fast convergence rates for minimizing both the gradient norm and function value, that is, $O(n \log \frac{1}{\epsilon} + \frac{n^{2/3}}{\epsilon^{2/3}})$ complexity for gradient norm and $O(n \log \frac{1}{\epsilon} + \sqrt{\frac{n}{\epsilon}})$ complexity for function value. Note that other stochastic approaches in Table 1 do not have this property.
- (Section 5) We propose an adaptively regularized accelerated SVRG variant, which does not require the knowledge of R_0 or Δ_0 and achieves a near-optimal complexity under IDC or IFC.

We put in extra efforts to make the proposed schemes as simple and elegant as possible. We believe that the simplicity makes the extensions of the new schemes easier.

2 Preliminaries

Throughout this paper, we use $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ to denote the inner product and the Euclidean norm, respectively. We let $[n]$ denote the set $\{1, 2, \ldots, n\}$, $\mathbb{E}$ denote the total expectation and $\mathbb{E}_{i_k}$ denote the expectation with respect to a random sample i_k. We say that a function $f : \mathbb{R}^d \to \mathbb{R}$ is *L-smooth* if it has L-Lipschitz continuous gradients, i.e.,

$$\forall x, y \in \mathbb{R}^d, \|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|.$$

A continuously differentiable f is called *μ-strongly convex* if

$$\forall x, y \in \mathbb{R}^d, f(x) - f(y) - \langle \nabla f(y), x - y \rangle \geq \frac{\mu}{2} \|x - y\|^2.$$

Other equivalent definitions of these two assumptions can be found in the textbook [44]. The following is an important consequence of a function f being L-smooth and convex:

$$\forall x, y \in \mathbb{R}^d, f(x) - f(y) - \langle \nabla f(y), x - y \rangle \geq \frac{1}{2L} \|\nabla f(x) - \nabla f(y)\|^2. \tag{1}$$

We call (1) the *interpolation condition* at (x, y) following [56]. If f is both L-smooth and μ-strongly convex, we can define a "shifted" function $h(x) = f(x) - f(x^\star) - \frac{\mu}{2} \|x - x^\star\|^2$ following [63]. It can be easily verified that h is $(L - \mu)$-smooth and convex, and thus from (1),

$$\forall x, y \in \mathbb{R}^d, h(x) - h(y) - \langle \nabla h(y), x - y \rangle \geq \frac{1}{2(L - \mu)} \|\nabla h(x) - \nabla h(y)\|^2, \tag{2}$$

which is equivalent to the *strongly convex interpolation condition* discovered in [56].

Oracle complexity (or simply complexity) refers to the required number of stochastic gradient ∇f_i computations to find an ϵ-accurate solution.

3 OGM-G: "Momentum" Reformulation and a Memory-Saving Variant

In this section, we focus on the IFC case, i.e., $f(x_0) - f(x^\star) \leq \Delta_0$. We use N to denote the total iteration number to prevent confusion (in other sections, we use K). Proofs in this section are given in

Algorithm 1 OGM-G: "Momentum" reformulation

Input: initial guess $x_0 \in \mathbb{R}^d$, total iteration number N.
Initialize: vector $v_0 = \mathbf{0}$, scalars $\theta_N = 1$ and $\theta_k^2 - \theta_k = \theta_{k+1}^2$, for $k = 0 \ldots N-1$.
for $k = 0, \ldots, N-1$ **do**
  $v_{k+1} = v_k + \frac{1}{L\theta_k \theta_{k+1}^2} \nabla f(x_k)$.
  $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k) - (2\theta_{k+1}^3 - \theta_{k+1}^2) v_{k+1}$.
end for
Output: x_N.

Appendix B. Recall that OGM-G has the following updates [33]. Let $y_0 = x_0$. For $k = 0, \ldots, N-1$,

$$
\begin{aligned}
y_{k+1} &= x_k - \frac{1}{L}\nabla f(x_k), \\
x_{k+1} &= y_{k+1} + \frac{(\theta_k - 1)(2\theta_{k+1} - 1)}{\theta_k(2\theta_k - 1)}(y_{k+1} - y_k) + \frac{2\theta_{k+1} - 1}{2\theta_k - 1}(y_{k+1} - x_k),
\end{aligned} \tag{3}
$$

where $\{\theta_k\}$ is recursively defined: $\theta_N = 1$ and $\begin{cases} \theta_k^2 - \theta_k = \theta_{k+1}^2 & k = 1 \ldots N-1, \\ \theta_0^2 - \theta_0 = 2\theta_1^2 & \text{otherwise.} \end{cases}$

OGM-G was discovered from the numerical solution to an SDP problem and its analysis is to show that the step coefficients in (3) specify a feasible solution to the SDP problem. While this analysis is natural for the PEP approach, it is hard to understand how each coefficient affects the rate, especially if one wants to generalize the scheme. Here we provide a simple algebraic analysis for OGM-G.

We start with a reformulation[3] of OGM-G in Algorithm 1, which aims to simplify the proof. We adopt a consistent $\{\theta_k\}$: $\theta_N = 1$ and $\theta_k^2 - \theta_k = \theta_{k+1}^2, k = 0 \ldots N-1$, which only costs a constant factor.[4] Interestingly, the reformulated scheme resembles the heavy-ball momentum method [49]. However, it can be shown that Algorithm 1 is not covered by the heavy-ball momentum scheme. Defining $\theta_{N+1}^2 = \theta_N^2 - \theta_N = 0$, we provide the one-iteration analysis in the following proposition:

Proposition 3.1. *In Algorithm 1, the following holds at any iteration* $k \in \{0, \ldots, N-1\}$:

$$
\begin{aligned}
A_k + B_{k+1} + C_{k+1} + E_{k+1} \le A_{k+1} + B_k + C_k + E_k - \theta_{k+1} \langle \nabla f(x_{k+1}), v_{k+1} \rangle \\
+ \sum_{i=k+1}^{N} \frac{\theta_i}{L\theta_k \theta_{k+1}^2} \langle \nabla f(x_k), \nabla f(x_i) \rangle,
\end{aligned} \tag{4}
$$

with $A_k \triangleq \frac{1}{\theta_k^2}(f(x_N) - f(x^\star) - \frac{1}{2L}\|\nabla f(x_N)\|^2)$, $B_k \triangleq \frac{1}{\theta_k^2}(f(x_k) - f(x^\star))$, $C_k \triangleq \frac{1}{2L\theta_k^2}\|\nabla f(x_k)\|^2$, $E_k \triangleq \frac{\theta_{k+1}^2}{\theta_k}\langle \nabla f(x_k), v_k \rangle$.

Remark 3.1.1. *A recent work [15] also conducted an algebraic analysis of OGM-G under a potential function framework. Their potential function decrease can be directly obtained from Proposition 3.1 by summing up* (4)*. By contrast, our "momentum" vector* $\{v_k\}$ *naturally merges into the analysis, which significantly simplifies the analysis. Moreover, it provides a better interpretation on how OGM-G utilizes the past gradients to achieve acceleration.*

From (4), we see that only the last two terms do not telescope. Note that the "momentum" vector is a weighted sum of the past gradients, i.e., $v_{k+1} = \sum_{i=0}^{k} \frac{1}{L\theta_i \theta_{i+1}^2}\nabla f(x_i)$. If we sum the terms up from $k = 0, \ldots, N-1$, it can be verified that they exactly sum up to 0. The presence of these special terms prevents OGM-G to have a usual potential function (e.g., those in [6]). Then, by telescoping the remaining terms, we obtain the final convergence guarantee.

Theorem 3.1. *The output of Algorithm 1 satisfies* $\|\nabla f(x_N)\|^2 \le \frac{8L\Delta_0}{(N+2)^2}$.

We observe two drawbacks of OGM-G (same as the algorithm description in [15]): (1) it requires storing a pre-computed parameter sequence, which costs $O(\frac{1}{\epsilon})$ floats; (2) except for the last iterate,

[3] It can be verified that this scheme is equivalent to the original one (3) through $v_k = \frac{1}{(2\theta_k - 1)\theta_k^2}(y_k - x_k)$.

[4] The original guarantee of OGM-G can be recovered if we set $\theta_0^2 - \theta_0 = 2\theta_1^2$.

Algorithm 2 M-OGM-G: Memory-saving OGM-G

Input: initial guess $x_0 \in \mathbb{R}^d$, total iteration number N.
Initialize: vector $v_0 = \mathbf{0}$.

for $k = 0, \dots, N-1$ **do**

$\quad v_{k+1} = v_k + \frac{12}{L(N-k+1)(N-k+2)(N-k+3)} \nabla f(x_k)$.

$\quad x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k) - \frac{(N-k)(N-k+1)(N-k+2)}{6} v_{k+1}$.

end for

Output: x_N or $\arg\min_{x \in \{x_0, \dots, x_N\}} \|\nabla f(x)\|$.

all other iterates are not known to have guarantees. We resolve these issues by proposing another parameterization of Algorithm 1 in the next subsection.

3.1 Memory-Saving OGM-G

A straightforward idea to resolve the aforementioned issues is to generalize Algorithm 1. However, we find it rather difficult since the parameters in the analysis are rather strict (despite that the proof is already simple). We choose to rely on computer-aided techniques [19]. The derivation of this variant (Algorithm 2) is based on the following numerical experiment.

Numerical experiment. OGM-G was discovered when considering the relaxed PEP problem [33]:

$$\max_{\substack{\nabla f(x_0), \dots, \nabla f(x_N) \in \mathbb{R}^d \\ f(x_0), \dots, f(x_N), f(x^\star) \in \mathbb{R}}} \|\nabla f(x_N)\|^2$$

$$\text{subject to} \begin{cases} \text{interpolation condition (1) at } (x_k, x_{k+1}), & k = 0, \dots, N-1, \\ \text{interpolation condition (1) at } (x_N, x_k), & k = 0, \dots, N-1, \\ \text{interpolation condition (1) at } (x_N, x^\star), & f(x_0) - f(x^\star) \le \Delta_0, \end{cases} \tag{P}$$

where the sequence $\{x_k\}$ is defined as $x_{k+1} = x_k - \frac{1}{L}\sum_{i=0}^{k} h_{k+1,i} \nabla f(x_i)$, $k = 0, \dots, N-1$ for some step coefficients $h \in \mathbb{R}^{N(N+1)/2}$. Given N, the step coefficients of OGM-G correspond to a numerical solution to the problem: $\arg\min_h \{\text{Lagrangian dual of (P)}\}$, which is denoted as (HD). Conceptually, solving problem (HD) would give us the fastest possible step coefficients under the constraints.[5] We expect there to be some constant-time slower schemes, which are neglected when solving (HD). To identify such schemes, we relax a set of interpolation conditions in problem (P):

$$f(x_N) - f(x_k) - \langle \nabla f(x_k), x_N - x_k \rangle \ge \frac{1}{2L} \|\nabla f(x_N) - \nabla f(x_k)\|^2 - \rho \|\nabla f(x_k)\|^2,$$

for $k = 0, \dots, N-1$ and some $\rho > 0$. After this relaxation, solving (HD) will no longer give us the step coefficients of OGM-G. By trying different ρ and checking the dependence on N, we discover Algorithm 2 when $\rho = \frac{1}{2L}$. Similar to our analysis of OGM-G, we provide a simple algebraic analysis for the new variant in the following theorem.

Theorem 3.2. *Define* $\delta_{k+1} \triangleq \frac{12}{(N-k+1)(N-k+2)(N-k+3)}$, $k = 0, \dots, N$. *In Algorithm 2, it holds that*

$$\sum_{k=0}^{N} \frac{\delta_{k+1}}{2} \|\nabla f(x_k)\|^2 \le \frac{12 L \Delta_0}{(N+2)(N+3)}. \tag{5}$$

Remark 3.2.1. *Algorithm 2 converges optimally on the last iterate (note that $\delta_{N+1} = 2$) and the minimum gradient since*

$$\min_{k \in \{0, \dots, N\}} \|\nabla f(x_k)\|^2 \le \frac{1}{\sum_{k=0}^{N} \frac{\delta_{k+1}}{2}} \sum_{k=0}^{N} \frac{\delta_{k+1}}{2} \|\nabla f(x_k)\|^2 \le \frac{8 L \Delta_0}{(N+2)(N+3) - 2}.$$

Clearly, the parameters of this variant can be computed on the fly and from (5), each iterate has a guarantee (although the guarantee degenerates quickly as $k \to 0$ since $1/\delta_{k+1} = \Omega((N-k)^3)$). Moreover, we can extend the benefits into the IDC case using the ideas in [42] as summarized below.

[5] However, since problem (HD) is non-convex, we can only obtain approximate solutions.

Algorithm 3 Acc-SVRG-G: Accelerated SVRG for Gradient minimization

Input: parameters $\{\tau_k\}$, $\{p_k\}$, initial guess $x_0 \in \mathbb{R}^d$, total iteration number K.
Initialize: vectors $z_0 = \tilde{x}_0 = x_0$ and scalars $\alpha_k = \frac{L\tau_k}{1-\tau_k}, \forall k$ and $\widetilde{\tau} = \sum_{k=0}^{K-1} \tau_k^{-2}$.

1: **for** $k = 0, \ldots, K-1$ **do**
2: $\quad y_k = \tau_k z_k + (1-\tau_k)\left(\tilde{x}_k - \frac{1}{L}\nabla f(\tilde{x}_k)\right)$.
3: $\quad z_{k+1} = \arg\min_x \left\{\langle \mathcal{G}_k, x\rangle + (\alpha_k/2)\left\|x - z_k\right\|^2\right\}$.
4: $\quad$ // $\mathcal{G}_k \triangleq \nabla f_{i_k}(y_k) - \nabla f_{i_k}(\tilde{x}_k) + \nabla f(\tilde{x}_k)$, where i_k is sampled uniformly in $[n]$.
5: $\quad \tilde{x}_{k+1} = \begin{cases} y_k & \text{with probability } p_k, \\ \tilde{x}_k & \text{with probability } 1 - p_k. \end{cases}$
6: **end for**

Output (for gradient): x_{out} is sampled from $\left\{\text{Prob}\{x_{\text{out}} = \tilde{x}_k\} = \frac{\tau_k^{-2}}{\widetilde{\tau}} \,\middle|\, k \in \{0, \ldots, K-1\}\right\}$.
Output (for function value): $\tilde{x}_K$.

Corollary 3.2.1 (IDC case)**.** *If we first run $N/2$ iterations of NAG and then continue with $N/2$ iterations of Algorithm 2, we obtain an output satisfying $\|\nabla f(x_N)\| = O(\frac{LR_0}{N^2})$.*

4 Accelerated SVRG: Fast Rates for Both Gradient Norm and Objective

In this section, we focus on the IDC case, i.e., $\|x_0 - x^\star\| \leq R_0$ for some $x^\star \in \mathcal{X}^\star$. From the development in the previous section, it is natural to ask whether we can use the PEP approach to motivate new stochastic schemes. However, due to the exponential growth of the number of possible states $(i_0, i_1, \ldots)$, we cannot directly adopt this approach. A feasible alternative is to first fix an algorithmic framework and a family of potential functions, and then use the potential-based PEP approach in [54]. However, this approach is much more restrictive. For example, it cannot identify special constructions like (4) in OGM-G. Fortunately, as we will see, we can get some inspiration from the recent development of deterministic methods. Proofs in this section are given in Appendix C.

Our proposed scheme is given in Algorithm 3. We adopt the elegant loopless design of SVRG in [34]. Note that the full gradient $\nabla f(\tilde{x}_k)$ is computed and stored only when $\tilde{x}_{k+1} = y_k$ at Step 5. We summarize our main technical novelty as follows.

Main algorithmic novelty. The design of stochastic accelerated methods is largely inspired by NAG. To make it clear, by setting $n = 1$, we see that Katyusha [1], MiG [61], SSNM [62], Varag [36], VRADA [52], ANITA [38], the acceleration framework in [16] and AC-SA [35, 24] all reduce to one of the following variants of NAG. We say that these methods are under the NAG framework.

$$\begin{cases} x_k = \tau_k z_k + (1-\tau_k) y_k, \\ z_{k+1} = z_k - \alpha_k \nabla f(x_k), \\ y_{k+1} = \tau_k z_{k+1} + (1-\tau_k) y_k. \end{cases}$$

Auslender and Teboulle [5]

$$\begin{cases} x_k = \tau_k z_k + (1-\tau_k) y_k, \\ z_{k+1} = z_k - \alpha_k \nabla f(x_k), \\ y_{k+1} = x_k - \eta_k \nabla f(x_k). \end{cases}$$

Linear Coupling [64]

See [57, 12] for other variants of NAG. When $n = 1$, Algorithm 3 reduces to the following scheme:

$$\begin{cases} y_k = \tau_k z_k + (1-\tau_k)\left(y_{k-1} - \frac{1}{L}\nabla f(y_{k-1})\right), \\ z_{k+1} = z_k - \frac{1}{\alpha_k}\nabla f(y_k). \end{cases}$$

Optimized Gradient Method (OGM) [19, 30]

Algorithm 3 reduces to the scheme of OGM when $n = 1$ (this point is clearer in the formulation of ITEM in [55]). OGM has a constant-time faster worst-case rate than NAG, which exactly matches the lower complexity bound established in [17]. In the following proposition, we show that the OGM framework helps us conduct a tight one-iteration analysis, which gives room for achieving our goal.

Proposition 4.1. *In Algorithm 3, the following holds at any iteration $k \geq 0$ and $\forall x^\star \in \mathcal{X}^\star$:*

$$\begin{aligned}
&\left(\frac{1-\tau_k}{\tau_k^2 p_k}\mathbb{E}\left[f(\tilde{x}_{k+1}) - f(x^\star)\right] + \frac{L}{2}\mathbb{E}\left[\left\|z_{k+1} - x^\star\right\|^2\right]\right) + \frac{(1-\tau_k)^2}{2L\tau_k^2}\mathbb{E}\left[\left\|\nabla f(\tilde{x}_k)\right\|^2\right] \\
&\leq \left(\frac{(1-\tau_k p_k)(1-\tau_k)}{\tau_k^2 p_k}\mathbb{E}\left[f(\tilde{x}_k) - f(x^\star)\right] + \frac{L}{2}\mathbb{E}\left[\left\|z_k - x^\star\right\|^2\right]\right).
\end{aligned} \tag{6}$$

The terms inside the parentheses form the commonly used potential function of SVRG variants. The additional $\mathbb{E}[\|\nabla f(\tilde{x}_k)\|^2]$ term is created by adopting the OGM framework. In other words, we use the following potential function for Algorithm 3 ($a_k, b_k, c_k \geq 0$):

$$T_k = a_k \mathbb{E}\left[f(\tilde{x}_k) - f(x^\star)\right] + b_k \mathbb{E}\left[\left\|z_k - x^\star\right\|^2\right] + \sum_{i=0}^{k-1} c_i \mathbb{E}\left[\left\|\nabla f(\tilde{x}_i)\right\|^2\right].$$

We first provide a simple parameter choice, which leads to a simple and clean analysis.

Theorem 4.1 (Single-stage parameter choice)**.** *In Algorithm 3, if we choose $p_k \equiv \frac{1}{n}, \tau_k = \frac{3}{k/n+6}$, then the following holds at the outputs:*

$$\begin{aligned}
\mathbb{E}\left[\left\|\nabla f(x_{\text{out}})\right\|^2\right] &= O\left(\frac{n^3 L\big(f(x_0) - f(x^\star)\big) + n^2 L^2 R_0^2}{K^3}\right), \\
\mathbb{E}\left[f(\tilde{x}_K)\right] - f(x^\star) &= O\left(\frac{n^2\big(f(x_0) - f(x^\star)\big) + nLR_0^2}{K^2}\right).
\end{aligned} \tag{7}$$

In other words, to guarantee that $\mathbb{E}\left[\|\nabla f(x_{\text{out}})\|\right] \leq \epsilon_g$ and $\mathbb{E}\left[f(\tilde{x}_K)\right] - f(x^\star) \leq \epsilon_f$, the oracle complexities are $O\Big(\frac{n(L(f(x_0)-f(x^\star)))^{1/3}}{\epsilon_g^{2/3}} + \frac{(nLR_0)^{2/3}}{\epsilon_g^{2/3}}\Big)$ and $O\Big(n\sqrt{\frac{f(x_0)-f(x^\star)}{\epsilon_f}} + \frac{\sqrt{nL}R_0}{\sqrt{\epsilon_f}}\Big)$, respectively.

From (7), we see that Algorithm 3 achieves fast $O(\frac{1}{K^{1.5}})$ and $O(\frac{1}{K^2})$ rates for minimizing the gradient norm and function value at the same time. However, despite being a simple choice, the oracle complexities are not better than the deterministic methods in Table 1. Below we provide a two-stage parameter choice, which is inspired by the idea of including a "warm-up phase" in [3, 36, 52, 38]. This theorem corresponds to the reported result in Table 1.

Theorem 4.2 (Two-stage parameter choice)**.** *In Algorithm 3, let $p_k = \max\{\frac{6}{k+8}, \frac{1}{n}\}, \tau_k = \frac{3}{p_k(k+8)}$. The oracle complexities needed to guarantee $\mathbb{E}\left[\|\nabla f(x_{\text{out}})\|\right] \leq \epsilon_g$ and $\mathbb{E}\left[f(\tilde{x}_K)\right] - f(x^\star) \leq \epsilon_f$ are*

$$O\left(n \min\left\{\log\frac{LR_0}{\epsilon_g}, \log n\right\} + \frac{(nLR_0)^{2/3}}{\epsilon_g^{2/3}}\right) \text{ and } O\left(n \min\left\{\log\frac{LR_0^2}{\epsilon_f}, \log n\right\} + \frac{\sqrt{nL}R_0}{\sqrt{\epsilon_f}}\right),$$

respectively.

If ϵ is large or n is very large, the recently proposed ANITA [38] achieves an $O(n)$ complexity, which matches the lower complexity bound $\Omega(n)$ in this case [58]. Since ANITA uses the NAG framework, we show that similar results can be derived under the OGM framework in the following theorem:

Theorem 4.3 (Low accuracy parameter choice)**.** *In Algorithm 3, let iteration N be the first time Step 5 updates $\tilde{x}_{k+1} = y_k$. If we choose $p_k \equiv \frac{1}{n}$, $\tau_k \equiv 1 - \frac{1}{\sqrt{n+1}}$ and terminate Algorithm 3 at iteration N, then the following holds at $\tilde{x}_{N+1}$:*

$$\mathbb{E}\left[\left\|\nabla f(\tilde{x}_{N+1})\right\|^2\right] \leq \frac{8L^2R_0^2}{5(\sqrt{n+1}+1)} \text{ and } \mathbb{E}\left[f(\tilde{x}_{N+1})\right] - f(x^\star) \leq \frac{LR_0^2}{\sqrt{n+1}+1}.$$

In particular, if the required accuracies are low (or n is very large), i.e., $\epsilon_g^2 \geq \frac{8L^2R_0^2}{5(\sqrt{n+1}+1)}$ and $\epsilon_f \geq \frac{LR_0^2}{\sqrt{n+1}+1}$, then Algorithm 3 only has an $O(n)$ oracle complexity.

In the low accuracy region (specified above), the choice in Theorem 4.3 removes the $O(\log\frac{1}{\epsilon})$ factor in the complexity of Theorem 4.2. We include some numerical justifications of Algorithm 3 in Appendix A. We believe that the potential-based PEP approach in [54] can help us identify better parameter choices of Algorithm 3, which we leave for future work.

Algorithm 4 R-Acc-SVRG-G

Input: accuracy $\epsilon > 0$, parameters $\delta_0 = L, \beta > 1$, initial guess $x_0 \in \mathbb{R}^d$.
1: **for** $t = 0, 1, 2, \ldots$ **do**
2: Define $f^{\delta_t}(x) = (1/n)\sum_{i=1}^n f_i^{\delta_t}(x)$, where $f_i^{\delta_t}(x) = f_i(x) + (\delta_t/2)\left\|x - x_0\right\|^2$.
3: Initialize vectors $z_0 = \tilde{x}_0 = x_0$ and set $\tau_x, \tau_z, \alpha, p, C_{\text{IDC}}, C_{\text{IFC}}$ according to Proposition 5.1.
4: **for** $k = 0, 1, 2, \ldots$ **do**
5: $y_k = \tau_x z_k + (1 - \tau_x)\tilde{x}_k + \tau_z\left(\delta_t(\tilde{x}_k - z_k) - \nabla f^{\delta_t}(\tilde{x}_k)\right)$.
6: $z_{k+1} = \arg\min_x \left\{ \left\langle \mathcal{G}_k^{\delta_t}, x\right\rangle + (\alpha/2)\left\|x - z_k\right\|^2 + (\delta_t/2)\left\|x - y_k\right\|^2 \right\}$.
7: // $\mathcal{G}_k^{\delta_t} \triangleq \nabla f_{i_k}^{\delta_t}(y_k) - \nabla f_{i_k}^{\delta_t}(\tilde{x}_k) + \nabla f^{\delta_t}(\tilde{x}_k)$, where i_k is sampled uniformly in $[n]$.
8: $\tilde{x}_{k+1} = \begin{cases} y_k & \text{with probability } p, \\ \tilde{x}_k & \text{with probability } 1 - p. \end{cases}$
9: **if** [6] $\left\|\nabla f(\tilde{x}_k)\right\| \leq \epsilon$ **then** output $\tilde{x}_k$ and terminate the algorithm.
10: **if** under IDC and $(1 + \frac{\delta_t}{\alpha})^k \geq \sqrt{C_{\text{IDC}}}/\delta_t$ **then** break the inner loop.
11: **if** under IFC and $(1 + \frac{\delta_t}{\alpha})^k \geq \sqrt{C_{\text{IFC}}/2\delta_t}$ **then** break the inner loop.
12: **end for**
13: $\delta_{t+1} = \delta_t/\beta$.
14: **end for**

5 Near-Optimal Accelerated SVRG with Adaptive Regularization

Currently, there is no known stochastic method that directly achieves the optimal rate in ϵ. To get near-optimal rates, the existing strategy is to use a carefully designed regularization technique [42, 2] with a method that solves strongly convex problems; see, e.g., [42, 2, 22, 11]. However, the regularization parameter requires the knowledge of R_0 or Δ_0, which significantly limits its practicality.

Inspired by the recently proposed adaptive regularization technique [27], we develop a near-optimal accelerated SVRG variant (Algorithm 4) that does not require the knowledge of R_0 or Δ_0. Note that this technique was originally proposed for NAG under the IDC assumption. Our development extends this technique to the stochastic setting, which brings an $O(\sqrt{n})$ rate improvement. Moreover, we consider both IFC and IDC cases. Proofs in this section are provided in Appendix D.

Detailed design. Algorithm 4 has a "guess-and-check" framework. In the outer loop, we first define the regularized objective f^{δ_t} using the current estimate of regularization parameter δ_t, and then we initialize an accelerated SVRG method (the inner loop) to solve the δ_t-strongly convex f^{δ_t}. If the inner loop breaks at Step 10 or 11, indicating the poor quality of the current estimate δ_t, δ_t will be divided by a fixed β. Thus, conceptually, we can adopt any method that solves strongly convex finite-sums at the optimal rate as the inner loop. However, since the constructions of Step 10 or 11 require some algorithm-dependent constants, we have to fix one method as the inner loop.

The inner loop we adopted is a loopless variant of BS-SVRG [63]. This is because (i) BS-SVRG is the fastest known accelerated SVRG variant (for ill-conditioned problems) and (ii) it has a simple scheme, especially after using the loopless construction [34]. However, its original guarantee is built upon $\{z_k\}$. Clearly, we cannot implement the stopping criterion (Step 9) on $\left\|\nabla f(z_k)\right\|$. Interestingly, we discover that its sequence $\{\tilde{x}_k\}$ works perfectly in our regularization framework, even if we can neither establish convergence on $f(\tilde{x}_k) - f(x^\star)$ nor on $\left\|\tilde{x}_k - x^\star\right\|^2$.[7] Moreover, we find that the loopless construction significantly simplifies the parameter constraints of BS-SVRG, which originally involves $\Theta(n)$th-order inequality. We provide the detailed parameter choice as follows:

Proposition 5.1 (Parameter choice). *In Algorithm 4, we set* $\tau_x = \frac{\alpha+\delta_t}{\alpha+L+\delta_t}, \tau_z = \frac{\tau_x}{\delta_t} - \frac{\alpha(1-\tau_x)}{\delta_t L}$ *and* $p = \frac{1}{n}$. *We set α as the (unique) positive root of the cubic equation* $\left(1 - \frac{p(\alpha+\delta_t)}{\alpha+L+\delta_t}\right)\left(1 + \frac{\delta_t}{\alpha}\right)^2 = 1$ *and specify* $C_{\text{IDC}} = L^2 + \frac{L\alpha^2 p}{L+(1-p)(\alpha+\delta_t)}, C_{\text{IFC}} = 2L + \frac{2L\alpha^2 p}{(L+(1-p)(\alpha+\delta_t))\delta_t}$. *Under these choices, we have* $\frac{\alpha}{\delta_t} = O\big(n + \sqrt{n(L/\delta_t + 1)}\big), C_{\text{IDC}} = O\big((L + \delta_t)^2\big)$, *and* $C_{\text{IFC}} = O(L)$.

[6] Note that we maintain the full gradient $\nabla f^{\delta_t}(\tilde{x}_k)$ and $\nabla f(\tilde{x}_k) = \nabla f^{\delta_t}(\tilde{x}_k) - \delta_t(\tilde{x}_k - x_0)$.
[7] It is due to the special potential function of BS-SVRG (see (27)), which does not contain these two terms.

Under the choices of τ_x and τ_z, the α above is the optimal choice in our analysis. Then, we can characterize the progress of the inner loop in the following proposition:

Proposition 5.2 (The inner loop of Algorithm 4)**.** *Using the parameters specified in Proposition 5.1, after running the inner loop (Step 4-12) of Algorithm 4 for k iterations, we can conclude that*

(i) under IDC, i.e., $\|x_0 - x^\star\| \le R_0$ for some $x^\star \in \mathcal{X}^\star$,

$$\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|\right] \le \left(\delta_t + \left(1 + \frac{\delta_t}{\alpha}\right)^{-k} \sqrt{C_{\text{IDC}}}\right) R_0,$$

(ii) under IFC, i.e., $f(x_0) - f(x^\star) \le \Delta_0$,

$$\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|\right] \le \left(\sqrt{2\delta_t} + \left(1 + \frac{\delta_t}{\alpha}\right)^{-k} \sqrt{C_{\text{IFC}}}\right) \sqrt{\Delta_0}.$$

The above results motivate the design of Step 10 and 11. For example, in the IDC case, when the inner loop breaks at Step 10, using *(i)* above, we obtain $\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|\right] \le 2\delta_t R_0$. Then, by discussing the relative size of δ_t and a certain constant, we can estimate the complexity of Algorithm 4. The same methodology is used for the IFC case.

Theorem 5.1 (IDC case)**.** *Denote $\delta_{\text{IDC}}^\star = \frac{\epsilon q}{2R_0}$ for some $q \in (0,1)$ and let the outer iteration $t = \ell$ be the first time*[8] *$\delta_\ell \le \delta_{\text{IDC}}^\star$. The following assertions hold:*

(i) At outer iteration ℓ, Algorithm 4 terminates with probability at least $1 - q$.[9]

(ii) The total expected oracle complexity of the $\ell + 1$ outer loops is

$$O\left(\left(n \log \frac{LR_0}{\epsilon q} + \sqrt{\frac{nLR_0}{\epsilon q}}\right) \log \frac{LR_0}{\epsilon q}\right).$$

Theorem 5.2 (IFC case)**.** *Denote $\delta_{\text{IFC}}^\star = \frac{\epsilon^2 q^2}{8\Delta_0}$ for some $q \in (0,1)$ and let the outer iteration $t = \ell$ be the first time $\delta_\ell \le \delta_{\text{IFC}}^\star$. The following assertions hold:*

(i) At outer iteration ℓ, Algorithm 4 terminates with probability at least $1 - q$.

(ii) The total expected oracle complexity of the $\ell + 1$ outer loops is

$$O\left(\left(n \log \frac{\sqrt{L\Delta_0}}{\epsilon q} + \frac{\sqrt{nL\Delta_0}}{\epsilon q}\right) \log \frac{\sqrt{L\Delta_0}}{\epsilon q}\right).$$

Compared with regularized Katyusha in Table 1, the adaptive regularization approach drops the need to estimate R_0 or Δ_0 at the cost of a mere $\log \frac{1}{\epsilon}$ factor in the non-dominant term (if ϵ is small).

6 Discussion

In this work, we proposed several simple and practical schemes that complement existing works (Table 1). Admittedly, the new schemes are currently only limited to the unconstrained Euclidean setting, because our techniques heavily rely on the interpolation conditions (1) and (2). On the other hand, methods such as OGM [30], TM [51] and ITEM [55, 10], which also rely on these conditions, are still not known to have their proximal variants. We list a few future directions as follows.

(1) It is not clear how to naturally connect the parameters of M-OGM-G (Algorithm 2) to OGM-G (Algorithm 1). The parameters of both algorithms seem to be quite restrictive and hardly generalizable due to the special construction in (4). Does there exist an optimal method for minimizing the gradient norm that has a proper potential function (at each iteration)?

(2) Is this new "momentum" in OGM-G beneficial for training neural nets? Other classic momentum schemes such as NAG [40] or heavy-ball momentum method [49] are extremely effective for this task [53], and they were also originally proposed for convex objectives.

(3) Can we directly accelerate SARAH (L2S)? By extending OGM-G? It seems that existing stochastic acceleration techniques fail to accelerate SARAH (or result in poor dependence on n as in [16]).

[8] We assume that ϵ is small such that $\max\{\delta_{\text{IDC}}^\star, \delta_{\text{IFC}}^\star\} \le \delta_0 = L$ for simplicity. In this case, $\ell > 0$.

[9] If Algorithm 4 does not terminate at outer iteration ℓ, it terminates at the next outer iteration with probability at least $1 - q/\beta$. That is, it terminates with higher and higher probability. The same goes for the IFC case.

References

[1] Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, 18(1):8194–8244, 2017. 2, 6, 26

[2] Z. Allen-Zhu. How to make the gradients small stochastically: Even faster convex and nonconvex sgd. In *Advances in Neural Information Processing Systems*, pages 1157–1167, 2018. 1, 2, 8

[3] Z. Allen-Zhu and Y. Yuan. Improved SVRG for Non-Strongly-Convex or Sum-of-Non-Convex Objectives. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1080–1089, 2016. 7

[4] Z. Allen-Zhu, Y. Li, R. M. de Oliveira, and A. Wigderson. Much Faster Algorithms for Matrix Scaling. In C. Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science*, pages 890–901, 2017. 1

[5] A. Auslender and M. Teboulle. Interior gradient and proximal methods for convex and conic optimization. *SIAM Journal on Optimization*, 16(3):697–725, 2006. 6

[6] N. Bansal and A. Gupta. Potential-Function Proofs for Gradient Methods. *Theory of Computing*, 15(4):1–32, 2019. 4

[7] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Lower bounds for finding stationary points ii: first-order methods. *Mathematical Programming*, 185(1-2), 2021. 1, 2, 3

[8] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. 13, 14

[9] M. B. Cohen, A. Madry, D. Tsipras, and A. Vladu. Matrix Scaling and Balancing via Box Constrained Newton's Method and Interior Point Methods. In *IEEE 58th Annual Symposium on Foundations of Computer Science*, pages 902–913. IEEE, 2017. 1

[10] A. d'Aspremont, D. Scieur, and A. Taylor. Acceleration methods. *arXiv preprint arXiv:2101.09545*, 2021. 9

[11] D. Davis and D. Drusvyatskiy. Complexity of finding near-stationary points of convex functions stochastically. *arXiv preprint arXiv:1802.08556*, 2018. 8

[12] A. Defazio. On the Curved Geometry of Accelerated Optimization. In *Advances in Neural Information Processing Systems*, volume 32, pages 1764–1773, 2019. 6

[13] A. Defazio, F. R. Bach, and S. 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. 14

[14] J. Diakonikolas and C. Guzmán. Complementary Composite Minimization, Small Gradients in General Norms, and Applications to Regression Problems. *arXiv preprint arXiv:2101.11041*, 2021. 1

[15] J. Diakonikolas and P. Wang. Potential Function-based Framework for Making the Gradients Small in Convex and Min-Max Optimization. *arXiv preprint arXiv:2101.12101*, 2021. 1, 4

[16] D. Driggs, M. J. Ehrhardt, and C.-B. Schönlieb. Accelerating variance-reduced stochastic gradient methods. *Mathematical Programming*, 2020. doi: 10.1007/s10107-020-01566-2. 6, 9

[17] Y. Drori. The exact information-based complexity of smooth convex minimization. *Journal of Complexity*, 39:1–16, 2017. 1, 6

[18] Y. Drori and A. Taylor. On the oracle complexity of smooth strongly convex minimization. *arXiv preprint arXiv:2101.09740*, 2021. 1

[19] Y. Drori and M. Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1-2):451–482, 2014. 1, 3, 5, 6

[20] D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`. 13, 14

[21] C. Fang, C. J. Li, Z. Lin, and T. Zhang. SPIDER: Near-Optimal Non-Convex Optimization via Stochastic Path-Integrated Differential Estimator. In *Advances in Neural Information Processing Systems*, pages 687–697, 2018. 1

[22] D. J. Foster, A. Sekhari, O. Shamir, N. Srebro, K. Sridharan, and B. Woodworth. The Complexity of Making the Gradient Small in Stochastic Convex Optimization. In *Proceedings of the Thirty-Second Conference on Learning Theory*, pages 1319–1345, 2019. 1, 2, 8

[23] R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping From Saddle Points — Online Stochastic Gradient for Tensor Decomposition. In *Proceedings of The 28th Conference on Learning Theory*, pages 797–842, 2015. 1

[24] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012. 6

[25] S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. 1

[26] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016. 1

[27] M. Ito and M. Fukuda. Nearly optimal first-order methods for convex optimization under gradient norm measure: An adaptive regularization approach. *Journal of Optimization Theory and Applications*, 188(3):770–804, 2021. 1, 2, 8

[28] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan. How to Escape Saddle Points Efficiently. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1724–1732, 2017. 1

[29] R. Johnson and T. Zhang. Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013. 3, 14

[30] D. Kim and J. A. Fessler. Optimized first-order methods for smooth convex minimization. *Mathematical Programming*, 159(1):81–107, 2016. 1, 6, 9

[31] D. Kim and J. A. Fessler. Another Look at the Fast Iterative Shrinkage/Thresholding Algorithm (FISTA). *SIAM Journal on Optimization*, 28(1):223–250, 2018. 1, 3

[32] D. Kim and J. A. Fessler. Generalizing the optimized gradient method for smooth convex minimization. *SIAM Journal on Optimization*, 28(2):1920–1950, 2018. 1, 2, 3

[33] D. Kim and J. A. Fessler. Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions. *Journal of Optimization Theory and Applications*, 188(1): 192–219, 2021. 1, 2, 3, 4, 5

[34] D. Kovalev, S. Horváth, and P. Richtárik. Don't jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop. In *Algorithmic Learning Theory*, pages 451–467. PMLR, 2020. 6, 8

[35] G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012. 6

[36] G. Lan, Z. Li, and Y. Zhou. A unified variance-reduced accelerated gradient method for convex optimization. In *Advances in Neural Information Processing Systems*, volume 32, pages 10462–10472, 2019. 6, 7

[37] B. Li, M. Ma, and G. B. Giannakis. On the Convergence of SARAH and Beyond. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 223–233, 2020. 2, 14, 27

[38] Z. Li. ANITA: An Optimal Loopless Accelerated Variance-Reduced Gradient Method. *arXiv preprint arXiv:2103.11333*, 2021. 6, 7

[39] Q. Lin and L. Xiao. An Adaptive Accelerated Proximal Gradient Method and its Homotopy Continuation for Sparse Optimization. In *Proceedings of the 31th International Conference on Machine Learning*, pages 73–81, 2014. 1

[40] Y. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983. 1, 2, 9

[41] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003. 1

[42] Y. Nesterov. How to make the gradients small. *Optima. Mathematical Optimization Society Newsletter*, (88):10–11, 2012. 1, 2, 5, 8, 22, 27

[43] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013. 1

[44] Y. Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018. 3, 18

[45] Y. Nesterov, A. Gasnikov, S. Guminov, and P. Dvurechensky. Primal–dual accelerated gradient methods with small-dimensional relaxation oracle. *Optimization Methods and Software*, pages 1–38, 2020. 2

[46] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč. SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2613–2621, 2017. 1, 2

[47] N. H. Pham, L. M. Nguyen, D. T. Phan, and Q. Tran-Dinh. ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization. *Journal of Machine Learning Research*, 21(110):1–48, 2020. 1

[48] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998. 14

[49] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964. 4, 9

[50] U. G. Rothblum and H. Schneider. Scalings of matrices which have prespecified row sums and column sums via optimization. *Linear Algebra and its Applications*, 114:737–764, 1989. 1

[51] B. V. Scoy, R. A. Freeman, and K. M. Lynch. The Fastest Known Globally Convergent First-Order Method for Minimizing Strongly Convex Functions. *IEEE Control Systems Letters*, 2(1): 49–54, 2017. 9

[52] C. Song, Y. Jiang, and Y. Ma. Variance Reduction via Accelerated Dual Averaging for Finite-Sum Optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 833–844, 2020. 6, 7

[53] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1139–1147, 2013. 9

[54] A. Taylor and F. Bach. Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions. In *Conference on Learning Theory*, pages 2934–2992, 2019. 2, 3, 6, 7

[55] A. Taylor and Y. Drori. An optimal gradient method for smooth strongly convex minimization. *arXiv preprint arXiv:2101.09741*, 2021. 1, 6, 9

[56] A. B. Taylor, J. M. Hendrickx, and F. Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 161(1-2):307–345, 2017. 3

[57] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. https://www.mit.edu/~dimitrib/PTseng/papers/apgm.pdf, 2008. Accessed May 1, 2020. 6

[58] B. E. Woodworth and N. Srebro. Tight Complexity Bounds for Optimizing Composite Objectives. In *Advances in Neural Information Processing Systems*, pages 3639–3647, 2016. 3, 7

[59] L. Xiao and T. Zhang. A Proximal Stochastic Gradient Method with Progressive Variance Reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014. 3, 14

[60] D. Zhou, P. Xu, and Q. Gu. Stochastic Nested Variance Reduction for Nonconvex Optimization. *Journal of Machine Learning Research*, 21:103:1–103:63, 2020. 1

[61] K. Zhou, F. Shang, and J. Cheng. A Simple Stochastic Variance Reduced Algorithm with Fast Convergence Rates. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5980–5989, 2018. 6

[62] K. Zhou, Q. Ding, F. Shang, J. Cheng, D. Li, and Z.-Q. Luo. Direct Acceleration of SAGA using Sampled Negative Momentum. In *Proceedings of the Twenty Second International Conference on Artificial Intelligence and Statistics*, pages 1602–1610, 2019. 6

[63] K. Zhou, A. M.-C. So, and J. Cheng. Boosting First-Order Methods by Shifting Objective: New Schemes with Faster Worst-Case Rates. In *Advances in Neural Information Processing Systems*, pages 15405–15416, 2020. 3, 8, 22

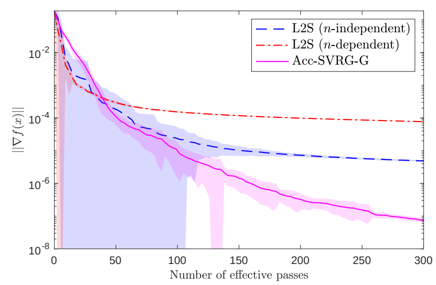
[64] Z. A. Zhu and L. Orecchia. Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent. In *8th Innovations in Theoretical Computer Science Conference*, volume 67 of *LIPIcs*, pages 3:1–3:22, 2017. 6

Checklist

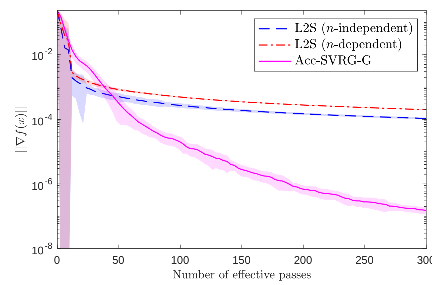
1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We are not aware of clear negative societal impacts since we focus on developing generic algorithms for convex optimization.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See the introduction.
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figure 1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Appendix A.
   (b) Did you mention the license of the assets? [Yes] LIBSVM [8] is under the BSD license.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Details can be found in the online dataset repositories [8, 20].
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Details can be found in the online dataset repositories [8, 20].
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Supplementary Materials for “Practical Schemes for Finding Near-Stationary Points of Convex Finite-Sums”

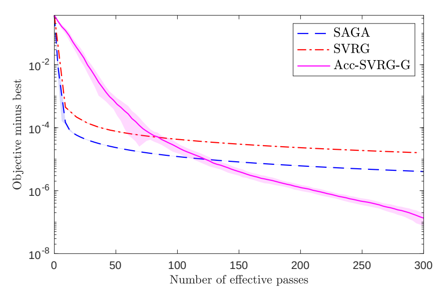
A Numerical results of Acc-SVRG-G (Algorithm 3)



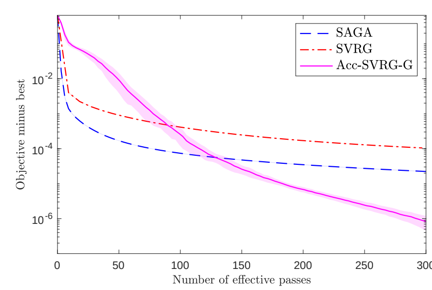
(a) a9a dataset. Measuring gradient norm.



(b) w8a dataset. Measuring gradient norm.



(c) a9a dataset. Measuring function value.



(d) w8a dataset. Measuring function value.

Figure 1: Performance evaluations. Run 20 seeds. Shaded bands indicate ± 1 standard deviation.

We did some experiments to justified the theoretical results (Theorem 4.2) of Acc-SVRG-G. We compared it to non-accelerated methods including L2S [37], SVRG [29, 59] and SAGA [13] under their original optimality measures. Note that other stochastic approaches in Table 1 require fixing the accuracy ϵ in advance, and thus are not convenient to be compared in the form of Figure 1. For measuring gradient norm, we simply tracked the smallest norm of all the full gradient computed to reduce complexity. Since the figures are in logarithmic scale, the deviation bands are asymmetric, and will emphasize the passes that have large deviations.

Setups. We ran the experiments on a Macbook Pro with a quad-core Intel Core i7-4870HQ with 2.50GHz cores, 16GB RAM, macOS Big Sur with Clang 12.0.5 and MATLAB R2020b. We were optimizing the binary logistic regression problem $f(x) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp\left(-b_i \langle a_i, x\rangle\right)\right)$ with dataset $a_i \in \mathbb{R}^d$, $b_i \in \{-1, +1\}$, $i \in [n]$. We used datasets from the LIBSVM website [8], including a9a [20] (32,561 samples, 123 features) and w8a [48] (49,749 samples, 300 features). We added one dimension as bias to all the datasets. We normalized the datasets and thus for this problem, $L = 0.25$. For Acc-SVRG-G, we chose the parameters according to Theorem 4.2. For L2S, we set $m = n$ and for its n-independent step size, we chose $\eta = \frac{c}{L}$ and tuned c using the same grid specified in [37]; for the n-dependent step size, we set $\eta = \frac{1}{L\sqrt{n}}$ according to Corollary 3 in [37]. For SAGA [13], we chose $\eta = \frac{1}{3L}$ following its theory. For SVRG [59], we set $\eta = \frac{1}{4L}$.

B Proofs of Section 3

To simplify the proof, we denote $D_k \triangleq f(x_k) - f(x^\star)$. And we use the following reformulation of interpolation condition (1) (at (x, y)) to facilitate our proof.

$$\forall x, y \in \mathbb{R}^d, \frac{1}{2L}\left(\|\nabla f(x)\|^2 + \|\nabla f(y)\|^2\right) + \left\langle \nabla f(y), x - y - \frac{1}{L}\nabla f(x) \right\rangle \le f(x) - f(y). \quad (8)$$

B.1 Proof to Proposition 3.1

We define $\theta_{N+1}^2 = \theta_N^2 - \theta_N = 0$. At iteration k, we are going to combine the reformulated interpolation conditions (8) at (x_k, x_{k+1}) and (x_N, x_k) with multipliers $\frac{1}{\theta_{k+1}^2}$ and $\frac{1}{\theta_k \theta_{k+1}^2}$, respectively.

$$\begin{aligned}
&\frac{1}{2L\theta_{k+1}^2}\left(\|\nabla f(x_k)\|^2 + \|\nabla f(x_{k+1})\|^2\right) + \frac{1}{\theta_{k+1}^2}\left\langle \nabla f(x_{k+1}), x_k - x_{k+1} - \frac{1}{L}\nabla f(x_k)\right\rangle \\
&\le \frac{1}{\theta_{k+1}^2}(D_k - D_{k+1}),
\end{aligned} \quad (9)$$

$$\begin{aligned}
&\frac{1}{2L\theta_k\theta_{k+1}^2}\left(\|\nabla f(x_N)\|^2 + \|\nabla f(x_k)\|^2\right) + \frac{1}{\theta_k\theta_{k+1}^2}\left\langle \nabla f(x_k), x_N - x_k - \frac{1}{L}\nabla f(x_N)\right\rangle \\
&\le \frac{1}{\theta_k\theta_{k+1}^2}(D_N - D_k).
\end{aligned} \quad (10)$$

Using the construction: $x_k - x_{k+1} = \frac{1}{L}\nabla f(x_k) + (2\theta_{k+1}^3 - \theta_{k+1}^2)v_{k+1}$, we can write (9) as

$$\begin{aligned}
&\frac{1}{2L\theta_{k+1}^2}\left(\|\nabla f(x_k)\|^2 + \|\nabla f(x_{k+1})\|^2\right) + (2\theta_{k+1} - 1)\left\langle \nabla f(x_{k+1}), v_{k+1}\right\rangle \\
&\le \frac{1}{\theta_{k+1}^2}(D_k - D_{k+1}).
\end{aligned} \quad (11)$$

Note that using $\theta_k^2 - \theta_k = \theta_{k+1}^2$, we have $2\theta_{k+1}^3 - \theta_{k+1}^2 = \theta_{k+1}^4 - \theta_{k+2}^4$. Then,

$$\begin{aligned}
x_k - x_N = \sum_{i=k}^{N-1}(x_i - x_{i+1}) &= \frac{1}{L}\sum_{i=k}^{N-1}\nabla f(x_i) + \sum_{i=k}^{N-1}(\theta_{i+1}^4 - \theta_{i+2}^4)v_{i+1} \\
&= \frac{1}{L}\sum_{i=k}^{N-1}\nabla f(x_i) + \theta_{k+1}^4 v_{k+1} + \sum_{i=k}^{N-2}\theta_{i+2}^4(v_{i+2} - v_{i+1}) \\
&\overset{(a)}{=} \frac{1}{L}\sum_{i=k}^{N-1}\nabla f(x_i) + \theta_{k+1}^4 v_{k+1} + \sum_{i=k}^{N-2}\frac{\theta_{i+2}^2}{L\theta_{i+1}}\nabla f(x_{i+1}) \\
&\overset{(b)}{=} \theta_{k+1}^4 v_k + \sum_{i=k}^{N-1}\frac{\theta_i}{L}\nabla f(x_i),
\end{aligned}$$

where (a) and (b) use the construction: $v_{k+1} = v_k + \frac{1}{L\theta_k\theta_{k+1}^2}\nabla f(x_k)$.

Thus, (10) can be written as

$$\begin{aligned}
\frac{1}{\theta_k\theta_{k+1}^2}(D_N - D_k) \ge \frac{1}{2L\theta_k\theta_{k+1}^2}\|\nabla f(x_N)\|^2 - \frac{\theta_k^2 + \theta_{k+1}^2}{2L\theta_k^2\theta_{k+1}^2}\|\nabla f(x_k)\|^2 \\
- \frac{\theta_{k+1}^2}{\theta_k}\left\langle \nabla f(x_k), v_k\right\rangle - \sum_{i=k+1}^{N}\frac{\theta_i}{L\theta_k\theta_{k+1}^2}\left\langle \nabla f(x_k), \nabla f(x_i)\right\rangle.
\end{aligned}$$

Summing this inequality and (11), and using the relation $\theta_k^2 - \theta_k = \theta_{k+1}^2$, we obtain

$$
\begin{aligned}
&\left(\frac{1}{\theta_{k+1}^2} - \frac{1}{\theta_k^2}\right)\left(D_N - \frac{1}{2L}\left\|\nabla f(x_N)\right\|^2\right) + \left(\frac{1}{\theta_k^2} D_k - \frac{1}{\theta_{k+1}^2} D_{k+1}\right) \\
\geq &\left(\frac{1}{2L\theta_{k+1}^2}\left\|\nabla f(x_{k+1})\right\|^2 - \frac{1}{2L\theta_k^2}\left\|\nabla f(x_k)\right\|^2\right) \\
&+ \left(\frac{\theta_{k+2}^2}{\theta_{k+1}}\left\langle \nabla f(x_{k+1}), v_{k+1}\right\rangle - \frac{\theta_{k+1}^2}{\theta_k}\left\langle \nabla f(x_k), v_k\right\rangle\right) \\
&\underbrace{+\, \theta_{k+1}\left\langle \nabla f(x_{k+1}), v_{k+1}\right\rangle - \sum_{i=k+1}^{N} \frac{\theta_i}{L\theta_k\theta_{k+1}^2}\left\langle \nabla f(x_k), \nabla f(x_i)\right\rangle}_{\mathcal{R}_1}.
\end{aligned}
\tag{12}
$$

B.2 Proof to Theorem 3.1

It is clear that except for $\mathcal{R}_1$, all terms in (12) telescope. Since $v_{k+1} = \sum_{i=0}^{k} \frac{1}{L\theta_i\theta_{i+1}^2}\nabla f(x_i)$, by defining a matrix $P \in \mathbb{R}^{(N+1)\times(N+1)}$ with $P_{ki} = \frac{\theta_k}{L\theta_i\theta_{i+1}^2}\left\langle \nabla f(x_k), \nabla f(x_i)\right\rangle$, we can write $\mathcal{R}_1$ as $\sum_{i=0}^{k} P_{(k+1)i} - \sum_{i=k+1}^{N} P_{ik}$. Summing these terms from $k = 0$ to $N-1$, we obtain

$$
\sum_{k=0}^{N-1}\sum_{i=0}^{k} P_{(k+1)i} - \sum_{k=0}^{N-1}\sum_{i=k+1}^{N} P_{ik} = \sum_{k=1}^{N}\sum_{i=0}^{k-1} P_{ki} - \sum_{i=0}^{N-1}\sum_{k=i+1}^{N} P_{ki} = 0.
$$

Both of the summations are equal to the sum of the lower triangular entries of P.

Then, telescoping (12) from $k = 0$ to $N-1$ (note that $v_0 = \mathbf{0}$), we obtain

$$
\left(1 - \frac{1}{\theta_0^2}\right)\left(D_N - \frac{1}{2L}\left\|\nabla f(x_N)\right\|^2\right) \geq D_N - \frac{1}{\theta_0^2} D_0 + \frac{1}{2L}\left\|\nabla f(x_N)\right\|^2 - \frac{1}{2L\theta_0^2}\left\|\nabla f(x_0)\right\|^2.
$$

Using $D_0 \geq \frac{1}{2L}\left\|\nabla f(x_0)\right\|^2$ and $D_N \geq \frac{1}{2L}\left\|\nabla f(x_N)\right\|^2$, we obtain

$$
\left\|\nabla f(x_N)\right\|^2 \leq \frac{2LD_0}{\theta_0^2}.
$$

Since $\theta_k = \frac{1+\sqrt{1+4\theta_{k+1}^2}}{2} \geq \frac{1}{2} + \theta_{k+1} \Rightarrow \theta_k \geq \frac{N-k}{2} + 1 \Rightarrow \theta_0 \geq \frac{N+2}{2}$, we have

$$
\left\|\nabla f(x_N)\right\|^2 \leq \frac{8L\big(f(x_0) - f(x^\star)\big)}{(N+2)^2}.
$$

B.3 Proof to Theorem 3.2

Define for $k = 0, \ldots, N$,

$$
\tau_k \triangleq \frac{(N-k+2)(N-k+3)}{6}, \quad \delta_{k+1} \triangleq \frac{12}{(N-k+1)(N-k+2)(N-k+3)} = \frac{1}{\tau_{k+1}} - \frac{1}{\tau_k}.
$$

At iteration k, we are going to combine the reformulated interpolation conditions (8) at (x_k, x_{k+1}) and (x_N, x_k) with multipliers $\frac{1}{\tau_{k+1}}$ and δ_{k+1}, respectively.

$$
\begin{aligned}
&\frac{1}{2L\tau_{k+1}}\left(\left\|\nabla f(x_k)\right\|^2 + \left\|\nabla f(x_{k+1})\right\|^2\right) + \frac{1}{\tau_{k+1}}\left\langle \nabla f(x_{k+1}), x_k - x_{k+1} - \frac{1}{L}\nabla f(x_k)\right\rangle \\
\leq &\frac{1}{\tau_{k+1}}(D_k - D_{k+1}),
\end{aligned}
\tag{13}
$$

$$
\begin{aligned}
&\frac{\delta_{k+1}}{2L}\left(\left\|\nabla f(x_N)\right\|^2 + \left\|\nabla f(x_k)\right\|^2\right) + \delta_{k+1}\left\langle \nabla f(x_k), x_N - x_k - \frac{1}{L}\nabla f(x_N)\right\rangle \\
\leq &\delta_{k+1}(D_N - D_k).
\end{aligned}
\tag{14}
$$

Note that from the construction of Algorithm 2,

$$x_k - x_{k+1} - \frac{1}{L}\nabla f(x_k) = \frac{(N-k)(N-k+1)(N-k+2)}{6} v_{k+1},$$

$$x_k - x_N = \sum_{i=k}^{N-1} \frac{1}{L}\nabla f(x_i) + \sum_{i=k}^{N-1} \frac{(N-i)(N-i+1)(N-i+2)}{6} v_{i+1}.$$

Thus, (13) can be written as

$$\frac{1}{2L\tau_{k+1}}\left(\|\nabla f(x_k)\|^2 + \|\nabla f(x_{k+1})\|^2\right) + (N-k)\langle \nabla f(x_{k+1}), v_{k+1}\rangle \le \frac{1}{\tau_{k+1}}(D_k - D_{k+1}). \quad (15)$$

Defining $\mathcal{Q}(j) \triangleq (j+3)(j+2)(j+1)j$, we have $\mathcal{Q}(j) - \mathcal{Q}(j-1) = 4j(j+1)(j+2)$. Then,

$$\begin{aligned}
x_k - x_N &= \sum_{i=k}^{N-1} \frac{1}{L}\nabla f(x_i) + \frac{1}{24}\sum_{i=k}^{N-1} (\mathcal{Q}(N-i) - \mathcal{Q}(N-i-1)) v_{i+1} \\
&= \sum_{i=k}^{N-1} \frac{1}{L}\nabla f(x_i) + \frac{1}{24}\left(\mathcal{Q}(N-k)v_{k+1} + \sum_{i=k+1}^{N-1} \mathcal{Q}(N-i)(v_{i+1} - v_i)\right) \\
&\overset{(a)}{=} \frac{\mathcal{Q}(N-k)}{24} v_{k+1} + \frac{1}{L}\nabla f(x_k) + \sum_{i=k+1}^{N-1} \frac{1}{L}\left(\frac{\mathcal{Q}(N-i)\delta_{i+1}}{24} + 1\right)\nabla f(x_i) \\
&\overset{(b)}{=} \frac{\mathcal{Q}(N-k)}{24} v_k + \sum_{i=k}^{N-1} \frac{N-i+2}{2L}\nabla f(x_i),
\end{aligned}$$

where (a) and (b) use the construction $v_{k+1} = v_k + \frac{\delta_{k+1}}{L}\nabla f(x_k)$.

Thus, (14) can be written as

$$\begin{aligned}
&\delta_{k+1}(D_N - D_k) \\
&\ge \frac{\delta_{k+1}}{2L}\left(\|\nabla f(x_N)\|^2 + \|\nabla f(x_k)\|^2\right) - \frac{N-k}{2}\langle \nabla f(x_k), v_k\rangle \\
&\quad - \frac{(N-k+2)\delta_{k+1}}{2L}\|\nabla f(x_k)\|^2 - \sum_{i=k+1}^{N} \frac{(N-i+2)\delta_{k+1}}{2L}\langle \nabla f(x_k), \nabla f(x_i)\rangle.
\end{aligned}$$

Summing the above inequality and (15), we obtain

$$\begin{aligned}
&\left(\frac{1}{\tau_{k+1}} - \frac{1}{\tau_k}\right)\left(D_N - \frac{1}{2L}\|\nabla f(x_N)\|^2\right) + \left(\frac{1}{\tau_k}D_k - \frac{1}{\tau_{k+1}}D_{k+1}\right) \\
&\ge \left(\frac{1}{2L\tau_{k+1}}\|\nabla f(x_{k+1})\|^2 - \frac{1}{2L\tau_k}\|\nabla f(x_k)\|^2\right) + \frac{\delta_{k+1}}{2L}\|\nabla f(x_k)\|^2 \\
&\quad + \left(\frac{N-k-1}{2}\langle \nabla f(x_{k+1}), v_{k+1}\rangle - \frac{N-k}{2}\langle \nabla f(x_k), v_k\rangle\right) \\
&\quad + \frac{N-k+1}{2}\langle \nabla f(x_{k+1}), v_{k+1}\rangle - \sum_{i=k+1}^{N} \frac{(N-i+2)\delta_{k+1}}{2L}\langle \nabla f(x_k), \nabla f(x_i)\rangle.
\end{aligned} \quad (16)$$

Since $v_{k+1} = \sum_{i=0}^{k} \frac{\delta_{i+1}}{L}\nabla f(x_i)$, the last two terms above have a similar structure as $\mathcal{R}_1$ at (12). Define a matrix $P \in \mathbb{R}^{(N+1)\times(N+1)}$ with $P_{ki} = \frac{(N-k+2)\delta_{i+1}}{2L}\langle \nabla f(x_k), \nabla f(x_i)\rangle$. The last two terms above can be written as $\sum_{i=0}^{k} P_{(k+1)i} - \sum_{i=k+1}^{N} P_{ik}$. If we sum these terms from $k = 0, \ldots, N-1$, they sum up to 0 (see Section B.2). Then, by telescoping (16) from $k = 0, \ldots, N-1$, we obtain

$$\begin{aligned}
&\frac{1}{2L}\|\nabla f(x_N)\|^2 - \frac{1}{2L\tau_0}\|\nabla f(x_0)\|^2 + \frac{1 - \frac{1}{\tau_0}}{2L}\|\nabla f(x_N)\|^2 + \sum_{k=0}^{N-1} \frac{\delta_{k+1}}{2L}\|\nabla f(x_k)\|^2 \\
&\le \left(1 - \frac{1}{\tau_0}\right)D_N + \frac{1}{\tau_0}D_0 - D_N.
\end{aligned}$$

Finally, using $D_0 \geq \frac{1}{2L}\left\|\nabla f(x_0)\right\|^2$ and $D_N \geq \frac{1}{2L}\left\|\nabla f(x_N)\right\|^2$, we obtain

$$\left\|\nabla f(x_N)\right\|^2 + \sum_{k=0}^{N-1} \frac{\delta_{k+1}}{2}\left\|\nabla f(x_k)\right\|^2 \leq \frac{2L}{\tau_0} D_0 = \frac{12L\big(f(x_0) - f(x^\star)\big)}{(N+2)(N+3)}. \tag{17}$$

B.4 Proof to Corollary 3.2.1

We assume N is divisible by 2 for simplicity. After running $N/2$ iterations of NAG, we obtain an output $x_{N/2}$ satisfying (cf. Theorem 2.2.2 in [44])

$$f(x_{N/2}) - f(x^\star) = O\left(\frac{LR_0^2}{N^2}\right).$$

Then, let $x_{N/2}$ be the input of Algorithm 2. Using (17), after running another $N/2$ iterations of Algorithm 2, we obtain

$$\left\|\nabla f(x_N)\right\|^2 = O\left(\frac{L^2R_0^2}{N^4}\right).$$

C Proofs of Section 4

C.1 Proof to Proposition 4.1

Using the interpolation condition (1) at $(x^\star, y_k)$, we obtain

$$\begin{aligned} f(y_k) - f(x^\star) \leq{}& \langle \nabla f(y_k), y_k - x^\star\rangle - \frac{1}{2L}\left\|\nabla f(y_k)\right\|^2 \\ \overset{(\star)}{\leq}{}& \frac{1-\tau_k}{\tau_k}\langle \nabla f(y_k), \tilde{x}_k - y_k\rangle - \frac{1-\tau_k}{L\tau_k}\langle \nabla f(y_k), \nabla f(\tilde{x}_k)\rangle \\ &+ \langle \nabla f(y_k), z_k - x^\star\rangle - \frac{1}{2L}\left\|\nabla f(y_k)\right\|^2, \end{aligned} \tag{18}$$

where $(\star)$ follows from the construction $y_k = \tau_k z_k + (1-\tau_k)\left(\tilde{x}_k - \frac{1}{L}\nabla f(\tilde{x}_k)\right)$.

From the optimality condition of Step 3, we can conclude that

$$\begin{aligned} &\mathcal{G}_k + \alpha_k(z_{k+1} - z_k) = \mathbf{0} \\ \overset{(a)}{\Rightarrow}\ & \langle \mathcal{G}_k, z_k - x^\star\rangle = \frac{1}{2\alpha_k}\left\|\mathcal{G}_k\right\|^2 + \frac{\alpha_k}{2}\left(\left\|z_k - x^\star\right\|^2 - \left\|z_{k+1} - x^\star\right\|^2\right) \\ \overset{(b)}{\Rightarrow}\ & \langle \nabla f(y_k), z_k - x^\star\rangle = \frac{1}{2\alpha_k}\mathbb{E}_{i_k}\left[\left\|\mathcal{G}_k\right\|^2\right] + \frac{\alpha_k}{2}\left(\left\|z_k - x^\star\right\|^2 - \mathbb{E}_{i_k}\left[\left\|z_{k+1} - x^\star\right\|^2\right]\right), \end{aligned} \tag{19}$$

where (a) uses $\langle u, v\rangle = \frac{1}{2}(\left\|u\right\|^2 + \left\|v\right\|^2 - \left\|u - v\right\|^2)$ and (b) follows from taking the expectation wrt sample i_k.

Using the interpolation condition (1) at $(\tilde{x}_k, y_k)$, we can bound $\mathbb{E}_{i_k}\left[\left\|\mathcal{G}_k\right\|^2\right]$ as

$$\begin{aligned} \mathbb{E}_{i_k}\left[\left\|\mathcal{G}_k\right\|^2\right] &= \mathbb{E}_{i_k}\left[\left\|\nabla f_{i_k}(y_k) - \nabla f_{i_k}(\tilde{x}_k)\right\|^2\right] + 2\langle \nabla f(y_k), \nabla f(\tilde{x}_k)\rangle - \left\|\nabla f(\tilde{x}_k)\right\|^2 \\ &\leq 2L\big(f(\tilde{x}_k) - f(y_k) - \langle \nabla f(y_k), \tilde{x}_k - y_k\rangle\big) + 2\langle \nabla f(y_k), \nabla f(\tilde{x}_k)\rangle \\ &\quad - \left\|\nabla f(\tilde{x}_k)\right\|^2. \end{aligned} \tag{20}$$

Combine (18), (19) and (20).

$$\begin{aligned} f(y_k) - f(x^\star) \leq{}& \frac{L}{\alpha_k}\big(f(\tilde{x}_k) - f(y_k)\big) + \left(\frac{1-\tau_k}{\tau_k} - \frac{L}{\alpha_k}\right)\langle \nabla f(y_k), \tilde{x}_k - y_k\rangle \\ &+ \left(\frac{1}{\alpha_k} - \frac{1-\tau_k}{L\tau_k}\right)\langle \nabla f(y_k), \nabla f(\tilde{x}_k)\rangle \\ &+ \frac{\alpha_k}{2}\left(\left\|z_k - x^\star\right\|^2 - \mathbb{E}_{i_k}\left[\left\|z_{k+1} - x^\star\right\|^2\right]\right) \\ &- \frac{1}{2L}\left\|\nabla f(y_k)\right\|^2 - \frac{1}{2\alpha_k}\left\|\nabla f(\tilde{x}_k)\right\|^2. \end{aligned}$$

Substitute the choice $\alpha_k = \frac{L\tau_k}{1-\tau_k}$.

$$\frac{1-\tau_k}{\tau_k^2}\big(f(y_k) - f(x^\star)\big) \leq \frac{(1-\tau_k)^2}{\tau_k^2}\big(f(\tilde{x}_k) - f(x^\star)\big) + \frac{L}{2}\left(\|z_k - x^\star\|^2 - \mathbb{E}_{i_k}\left[\|z_{k+1} - x^\star\|^2\right]\right)$$
$$- \frac{1-\tau_k}{2L\tau_k}\|\nabla f(y_k)\|^2 - \frac{(1-\tau_k)^2}{2L\tau_k^2}\|\nabla f(\tilde{x}_k)\|^2. \tag{21}$$

Note that by construction, $\mathbb{E}\left[f(\tilde{x}_{k+1})\right] = p_k \mathbb{E}\left[f(y_k)\right] + (1-p_k)\mathbb{E}\left[f(\tilde{x}_k)\right]$, and thus

$$\begin{aligned}\frac{1-\tau_k}{\tau_k^2 p_k}\mathbb{E}\left[f(\tilde{x}_{k+1}) - f(x^\star)\right] \leq{}& \frac{(1-\tau_k p_k)(1-\tau_k)}{\tau_k^2 p_k}\mathbb{E}\left[f(\tilde{x}_k) - f(x^\star)\right] \\ &+ \frac{L}{2}\left(\mathbb{E}\left[\|z_k - x^\star\|^2\right] - \mathbb{E}\left[\|z_{k+1} - x^\star\|^2\right]\right) \\ &- \frac{1-\tau_k}{2L\tau_k}\mathbb{E}\left[\|\nabla f(y_k)\|^2\right] - \frac{(1-\tau_k)^2}{2L\tau_k^2}\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|^2\right].\end{aligned}$$

C.2 Proof to Theorem 4.1

It can be easily verified that under this choice ($p_k \equiv \frac{1}{n}, \tau_k = \frac{3}{k/n+6}$), for any $k \geq 0, n \geq 1$,

$$\frac{(1-\tau_{k+1}p_{k+1})(1-\tau_{k+1})}{\tau_{k+1}^2 p_{k+1}} \leq \frac{1-\tau_k}{\tau_k^2 p_k}.$$

Then, using Proposition 4.1, after summing (6) from $k = 0, \ldots, K-1$, we obtain

$$\begin{aligned}&\frac{n(1-\tau_{K-1})}{\tau_{K-1}^2}\mathbb{E}\left[f(\tilde{x}_K) - f(x^\star)\right] + \frac{L}{2}\mathbb{E}\left[\|z_K - x^\star\|^2\right] + \sum_{k=0}^{K-1}\frac{(1-\tau_k)^2}{2L\tau_k^2}\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|^2\right] \\ &\leq (2n-1)\big(f(x_0) - f(x^\star)\big) + \frac{L}{2}\|x_0 - x^\star\|^2.\end{aligned}$$

Note that $\tau_k \leq \frac{1}{2}, \forall k$. We have the following two consequences of the above inequality.

$$\mathbb{E}\left[f(\tilde{x}_K)\right] - f(x^\star) \leq \tau_{K-1}^2\left(4\big(f(x_0) - f(x^\star)\big) + \frac{L}{n}\|x_0 - x^\star\|^2\right),$$

$$\begin{aligned}\mathbb{E}\left[\|\nabla f(x_{\text{out}})\|^2\right] &= \frac{1}{\sum_{k=0}^{K-1}\tau_k^{-2}}\sum_{k=0}^{K-1}\frac{1}{\tau_k^2}\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|^2\right] \\ &\leq \frac{16nL\big(f(x_0) - f(x^\star)\big) + 4L^2\|x_0 - x^\star\|^2}{\sum_{k=0}^{K-1}\tau_k^{-2}}.\end{aligned}$$

Substituting the parameter choice, we obtain

$$\mathbb{E}\left[f(\tilde{x}_K)\right] - f(x^\star) \leq \frac{36n^2\big(f(x_0) - f(x^\star)\big) + 9nL\|x_0 - x^\star\|^2}{(K+6n-1)^2} = \epsilon_f,$$

$$\mathbb{E}\left[\|\nabla f(x_{\text{out}})\|^2\right] \leq \frac{144nL\big(f(x_0) - f(x^\star)\big) + 36L^2\|x_0 - x^\star\|^2}{\sum_{k=0}^{K-1}\left(\frac{k}{n} + 6\right)^2}.$$

Note that

$$\sum_{k=0}^{K-1}\left(\frac{k}{n} + 6\right)^2 \geq \int_0^K\left(\frac{x-1}{n} + 6\right)^2 dx = \frac{(K+6n-1)^3 - (6n-1)^3}{3n^2}.$$

Thus,

$$\mathbb{E}\left[\|\nabla f(x_{\text{out}})\|\right]^2 \leq \mathbb{E}\left[\|\nabla f(x_{\text{out}})\|^2\right] \leq \frac{432n^3L\big(f(x_0) - f(x^\star)\big) + 108n^2L^2\|x_0 - x^\star\|^2}{(K+6n-1)^3 - (6n-1)^3} = \epsilon_g^2.$$

Since the expected iteration cost of Algorithm 3 is $\mathbb{E}\left[\#\text{grad}_k\right] = p_k(n+2) + (1-p_k)2 = 3$, to guarantee $\mathbb{E}\left[\|\nabla f(x_{\text{out}})\|\right] \le \epsilon_g$ and $\mathbb{E}\left[f(\tilde{x}_K)\right] - f(x^\star) \le \epsilon_f$, the total oracle complexities are $O\left(\frac{n(L(f(x_0)-f(x^\star)))^{1/3}}{\epsilon_g^{2/3}} + \frac{(nLR_0)^{2/3}}{\epsilon_g^{2/3}}\right)$ and $O\left(n\sqrt{\frac{f(x_0)-f(x^\star)}{\epsilon_f}} + \frac{\sqrt{n}LR_0}{\sqrt{\epsilon_f}}\right)$, respectively.

C.3 Proof to Theorem 4.2

First, it can be verified that for any $k \ge 0, n \ge 1$, the following inequality holds.

$$\frac{(1-\tau_{k+1}p_{k+1})(1-\tau_{k+1})}{\tau_{k+1}^2 p_{k+1}} \le \frac{1-\tau_k}{\tau_k^2 p_k}.$$

The verification can be done by considering the two cases: (i) $k+8 < 6n$, where $p_k = \frac{6}{k+8}, \tau_k = \frac{1}{2}$, (ii) $k+8 \ge 6n$, in which $p_k = \frac{1}{n}, \tau_k = \frac{3n}{k+8}$.

Then, using Proposition 4.1, after summing (6) from $k = 0, \ldots, K-1$, we obtain

$$\begin{aligned}
&\frac{1-\tau_{K-1}}{\tau_{K-1}^2 p_{K-1}}\mathbb{E}\left[f(\tilde{x}_K) - f(x^\star)\right] + \frac{L}{2}\mathbb{E}\left[\|z_K - x^\star\|^2\right] + \sum_{k=0}^{K-1}\frac{(1-\tau_k)^2}{2L\tau_k^2}\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|^2\right] \\
&\le \frac{5}{3}\big(f(x_0) - f(x^\star)\big) + \frac{L}{2}\|x_0 - x^\star\|^2 \le \frac{4}{3}LR_0^2.
\end{aligned}$$

Note that $\tau_k \le \frac{1}{2}, \forall k$. We can conclude the following two consequences.

$$\mathbb{E}\left[f(\tilde{x}_K)\right] - f(x^\star) \le \frac{8}{3}\tau_{K-1}^2 p_{K-1} LR_0^2, \tag{22}$$

$$\mathbb{E}\left[\|\nabla f(x_{\text{out}})\|^2\right] = \frac{1}{\sum_{k=0}^{K-1}\tau_k^{-2}}\sum_{k=0}^{K-1}\frac{1}{\tau_k^2}\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|^2\right] \le \frac{32L^2R_0^2}{3\sum_{k=0}^{K-1}\tau_k^{-2}}. \tag{23}$$

Now we consider two stages.

Stage I (low accuracy stage): $K+8 \le 6n$. In this stage, let the accuracies be $\epsilon_g^2 = \frac{8L^2R_0^2}{3K} \ge \frac{8L^2R_0^2}{3(6n-8)}$ and $\epsilon_f = \frac{4LR_0^2}{K+7} \ge \frac{4LR_0^2}{6n-1}$. By substituting the parameter choice, (22) and (23) can be written as

$$\begin{aligned}
\mathbb{E}\left[f(\tilde{x}_K)\right] - f(x^\star) &\le \frac{4LR_0^2}{K+7} = \epsilon_f, \\
\mathbb{E}\left[\|\nabla f(x_{\text{out}})\|\right]^2 \le \mathbb{E}\left[\|\nabla f(x_{\text{out}})\|^2\right] &\le \frac{8L^2R_0^2}{3K} = \epsilon_g^2.
\end{aligned}$$

Note that the expected iteration cost of Algorithm 3 is $\mathbb{E}\left[\#\text{grad}_k\right] = p_k(n+2) + (1-p_k)2 = np_k + 2$, and thus the total complexity in this stage is

$$\sum_{k=0}^{K-1}\mathbb{E}\left[\#\text{grad}_k\right] = n\sum_{k=0}^{K-1}\frac{6}{k+8} + 2K \le 6n\log(K+7) + 12n = O(n\log K).$$

Thus, the expected oracle complexities in this stage are $O(n\log\frac{LR_0}{\epsilon_g})$ and $O(n\log\frac{LR_0^2}{\epsilon_f})$, respectively.

Stage II (high accuracy stage): $K+8 > 6n$. In this stage, Algorithm 3 proceeds to find highly accurate solutions (i.e., $\epsilon_g^2 < \frac{8L^2R_0^2}{3(6n-8)}$ and $\epsilon_f < \frac{4LR_0^2}{6n-1}$). Substituting the parameter choice, we can write (22) and (23) as

$$\mathbb{E}\left[f(\tilde{x}_K)\right] - f(x^\star) \le \frac{24nLR_0^2}{(K+7)^2} = \epsilon_f, \tag{24}$$

$$\mathbb{E}\left[\|\nabla f(x_{\text{out}})\|^2\right] \le \frac{32L^2R_0^2}{3\left(24n - 28 + \sum_{k=6n-7}^{K-1}\tau_k^{-2}\right)} \overset{(\star)}{\le} \frac{288n^2L^2R_0^2}{(K+7)^3 + 432n^3 - 756n^2} = \epsilon_g^2, \tag{25}$$

where $(\star)$ follows from

$$\sum_{k=6n-7}^{K-1} \tau_k^{-2} = \frac{1}{9n^2} \sum_{k=6n-7}^{K-1} (k+8)^2 \geq \frac{1}{9n^2} \int_{6n-7}^{K} (x+7)^2 dx = \frac{(K+7)^3}{27n^2} - 8n.$$

Then, we count the expected complexity in this stage.

$$\sum_{k=0}^{K-1} \mathbb{E}\left[\#\text{grad}_k\right] = n \left(\sum_{k=0}^{6n-8} \frac{6}{k+8} + \sum_{k=6n-7}^{K-1} \frac{1}{n} \right) + 2K \leq 6n \log(6n) + 3K - 6n + 7.$$

Finally, combining with (24) and (25), we can conclude that the total expected oracle complexities in this stage are $O\Big(n \log n + \frac{(nLR_0)^{2/3}}{\epsilon_g^{2/3}}\Big)$ and $O\Big(n \log n + \frac{\sqrt{nL}R_0}{\sqrt{\epsilon_f}}\Big)$, respectively.

C.4 Proof to Theorem 4.3

We start at inequality (21) in the proof of Proposition 4.1, which is the consequence of one iteration k in Algorithm 3. Due to the constant choice of $\tau_k \equiv \tau$, we have

$$\begin{aligned} f(y_k) - f(x^\star) \leq (1-\tau)\big(f(\tilde{x}_k) - f(x^\star)\big) + \frac{L\tau^2}{2(1-\tau)} \left(\|z_k - x^\star\|^2 - \mathbb{E}_{i_k}\left[\|z_{k+1} - x^\star\|^2\right] \right) \\ - \frac{\tau}{2L} \|\nabla f(y_k)\|^2 - \frac{1-\tau}{2L} \|\nabla f(\tilde{x}_k)\|^2 . \end{aligned}$$

Since we fix $p_k \equiv p$ as a constant and terminate Algorithm 3 at the first time $\tilde{x}_{k+1} = y_k$ (denoted as the iteration N), it is clear that the random variable N follows the geometric distribution with parameter p, that is, for $k = 0, 1, 2, \ldots$, Prob $\{N = k\} = (1-p)^k p$. Moreover, since we have $\tilde{x}_N = \tilde{x}_{N-1} = \cdots = \tilde{x}_0 = x_0$, using the above inequality at iteration N, we obtain

$$\begin{aligned} \mathbb{E}\left[f(\tilde{x}_{N+1})\right] - f(x^\star) &\leq (1-\tau)\big(f(x_0) - f(x^\star)\big) + \frac{L\tau^2}{2(1-\tau)} \left(\mathbb{E}\left[\|z_N - x^\star\|^2 - \|z_{N+1} - x^\star\|^2\right] \right) \\ &\quad - \frac{\tau}{2L} \mathbb{E}\left[\|\nabla f(\tilde{x}_{N+1})\|^2\right] - \frac{1-\tau}{2L} \|\nabla f(x_0)\|^2 \\ &\overset{(\star)}{=} (1-\tau)\big(f(x_0) - f(x^\star)\big) + \frac{L\tau^2 p}{2(1-\tau)} \left(\|x_0 - x^\star\|^2 - \mathbb{E}\left[\|z_{N+1} - x^\star\|^2\right] \right) \\ &\quad - \frac{\tau}{2L} \mathbb{E}\left[\|\nabla f(\tilde{x}_{N+1})\|^2\right] - \frac{1-\tau}{2L} \|\nabla f(x_0)\|^2 , \end{aligned}$$

where $(\star)$ follows from

$$\begin{aligned} \mathbb{E}\left[\|z_{N+1} - x^\star\|^2\right] &= \frac{1}{1-p} \left(\sum_{k=0}^{\infty} (1-p)^k p \mathbb{E}\left[\|z_k - x^\star\|^2\right] - p \|z_0 - x^\star\|^2 \right) \\ &= \frac{1}{1-p} \left(\mathbb{E}\left[\|z_N - x^\star\|^2\right] - p \|z_0 - x^\star\|^2 \right). \end{aligned}$$

Thus, we can conclude that

$$\mathbb{E}\left[f(\tilde{x}_{N+1})\right] - f(x^\star) + \frac{\tau}{2L} \mathbb{E}\left[\|\nabla f(\tilde{x}_{N+1})\|^2\right] \leq \frac{L}{2} \left(1 - \tau + \frac{\tau^2 p}{1-\tau} \right) R_0^2.$$

Note that $\mathbb{E}[N] = \frac{1-p}{p}$ and the total expected oracle complexity is $n + 2(\mathbb{E}[N] + 1) = n + \frac{2}{p}$. We choose $p = \frac{1}{n}$, which leads to an $O(n)$ expected complexity. And we choose τ by minimizing the ratio $\Big(1 - \tau + \frac{\tau^2 p}{1-\tau}\Big)$ wrt τ. This gives $\tau = 1 - \frac{1}{\sqrt{n+1}} \geq \frac{1}{4}$ and

$$\mathbb{E}\left[f(\tilde{x}_{N+1})\right] - f(x^\star) + \frac{1}{8L} \mathbb{E}\left[\|\nabla f(\tilde{x}_{N+1})\|^2\right] \leq \frac{LR_0^2}{\sqrt{n+1} + 1}.$$

D Proofs of Section 5

We analyze Algorithm 4 following the "shifting" methodology in [63], which explores the tight interpolation condition (2) and leads to a simple and clean proof.

Note that after the regularization at Step 2, each $f_i^{\delta_t}$ is $(L+\delta_t)$-smooth and δ_t-strongly convex. We denote $x_{\delta_t}^\star$ as the unique minimizer of $\min_x f^{\delta_t}(x)$. Following [63], we define a "shifted" version of this problem: $\min_x h^{\delta_t}(x) = \frac{1}{n}\sum_{i=1}^n h_i^{\delta_t}(x)$, where

$$h_i^{\delta_t}(x) = f_i^{\delta_t}(x) - f_i^{\delta_t}(x_{\delta_t}^\star) - \left\langle \nabla f_i^{\delta_t}(x_{\delta_t}^\star), x - x_{\delta_t}^\star \right\rangle - \frac{\delta_t}{2}\left\| x - x_{\delta_t}^\star \right\|^2, \forall i.$$

It can be easily verified that each $h_i^{\delta_t}$ is L-smooth and convex. Note that $h_i^{\delta_t}(x_{\delta_t}^\star) = h^{\delta_t}(x_{\delta_t}^\star) = 0$ and $\nabla h_i^{\delta_t}(x_{\delta_t}^\star) = \nabla h^{\delta_t}(x_{\delta_t}^\star) = \mathbf{0}$, which means that h^{δ_t} and f^{δ_t} share the same minimizer $x_{\delta_t}^\star$.

Then, conceptually, we attempts to solve the "shifted" problem using an "shifted" SVRG gradient estimator: $\mathcal{H}_k^{\delta_t} \triangleq \nabla h_{i_k}^{\delta_t}(y_k) - \nabla h_{i_k}^{\delta_t}(\tilde{x}_k) + \nabla h^{\delta_t}(\tilde{x}_k)$. Clearly, the gradient of h^{δ_t} is not accessible due to the unknown $x_{\delta_t}^\star$. Zhou et al. [63] proposed a technical lemma (Lemma 1 below) to bypass this issue. Since the relation $\mathcal{H}_k^{\delta_t} = \mathcal{G}_k^{\delta_t} - \delta_t(y_k - x_{\delta_t}^\star)$ holds, we can use Lemma 1 as an instantiation of the "shifted" gradient oracle, see [63] for details.

D.1 Technical Lemmas

Lemma 1 (Lemma 1 in [63], the "shifting" technique)**.** *Given a gradient estimator $\mathcal{G}_y$ and vectors $z^+, z^-, y, x^\star \in \mathbb{R}^d$, fix the updating rule $z^+ = \arg\min_x \left\{ \langle \mathcal{G}_y, x\rangle + \frac{\alpha}{2}\left\|x - z^-\right\|^2 + \frac{\delta}{2}\left\|x - y\right\|^2 \right\}$. Suppose that we have a shifted gradient estimator $\mathcal{H}_y$ satisfying the relation $\mathcal{H}_y = \mathcal{G}_y - \delta(y - x^\star)$, it holds that*

$$\left\langle \mathcal{H}_y, z^- - x^\star \right\rangle = \frac{\alpha}{2}\left(\left\|z^- - x^\star\right\|^2 - \left(1 + \frac{\delta}{\alpha}\right)^2 \left\|z^+ - x^\star\right\|^2 \right) + \frac{1}{2\alpha}\left\|\mathcal{H}_y\right\|^2.$$

Lemma 2 (The regularization technique [42])**.** *For an L-smooth and convex function f and $\delta > 0$, defining $f^\delta(x) = f(x) + \frac{\delta}{2}\left\|x - x_0\right\|^2, \forall x$ and denoting $x_\delta^\star$ as the unique minimizer of f^δ, we have*

- *(i) f^δ is $(L+\delta)$-smooth and δ-strongly convex.*
- *(ii) $f^\delta(x_0) - f^\delta(x_\delta^\star) \le f(x_0) - f(x^\star)$.*
- *(iii) $\left\|x_0 - x_\delta^\star\right\|^2 \le \left\|x_0 - x^\star\right\|^2, \forall x^\star \in \mathcal{X}^\star$.*
- *(iv) $\left\|x_0 - x_\delta^\star\right\|^2 \le \frac{2}{\delta}\big(f(x_0) - f(x^\star)\big)$.*

Proof. *(i)* can be easily checked by the definition of L-smoothness and strong convexity. *(ii)* follows from $f^\delta(x_0) = f(x_0)$ and $f^\delta(x_\delta^\star) \ge f(x_\delta^\star) \ge f(x^\star)$. For *(iii)*, using the strong convexity of f^δ at $(x^\star, x_\delta^\star), \forall x^\star \in \mathcal{X}^\star$, we obtain

$$\begin{aligned} & f^\delta(x^\star) - f^\delta(x_\delta^\star) \ge \frac{\delta}{2}\left\|x^\star - x_\delta^\star\right\|^2 \\ \Rightarrow\ & f(x^\star) + \frac{\delta}{2}\left\|x^\star - x_0\right\|^2 - f(x_\delta^\star) - \frac{\delta}{2}\left\|x_\delta^\star - x_0\right\|^2 \ge \frac{\delta}{2}\left\|x^\star - x_\delta^\star\right\|^2 \\ \Rightarrow\ & \frac{\delta}{2}\left\|x_0 - x^\star\right\|^2 - \big(f(x_\delta^\star) - f(x^\star)\big) \ge \frac{\delta}{2}\left\|x_0 - x_\delta^\star\right\|^2 + \frac{\delta}{2}\left\|x^\star - x_\delta^\star\right\|^2. \end{aligned}$$

Then *(iii)* follows from the non-negativeness of $f(x_\delta^\star) - f(x^\star)$ and $\left\|x^\star - x_\delta^\star\right\|^2$. For *(iv)*, using the strong convexity of f^δ at $(x_0, x_\delta^\star)$ and *(ii)*, we have $\left\|x_0 - x_\delta^\star\right\|^2 \le \frac{2}{\delta}\big(f^\delta(x_0) - f^\delta(x_\delta^\star)\big) \le \frac{2}{\delta}\big(f(x_0) - f(x^\star)\big)$. □

D.2 Proof to Proposition 5.1

Denoting $\kappa_t = \frac{L+\delta_t}{\delta_t}$, we can write the equation $\left(1 - \frac{p(\alpha+\delta_t)}{\alpha+L+\delta_t}\right)\left(1 + \frac{\delta_t}{\alpha}\right)^2 = 1$ as

$$s\left(\frac{\alpha}{\delta_t}\right) \triangleq \left(\frac{\alpha}{\delta_t}\right)^3 - (2n-3)\left(\frac{\alpha}{\delta_t}\right)^2 - (2n\kappa_t + n - 3)\left(\frac{\alpha}{\delta_t}\right) - n\kappa_t + 1 = 0.$$

It can be verified that $s(2n+2\sqrt{n\kappa_t})>0$ for any $n\geq 1, \kappa_t>1$. Since $s(0)<0$ and $s(\frac{\alpha}{\delta_t})\to\infty$ as $\frac{\alpha}{\delta_t}\to\infty$, the unique positive root satisfies $\frac{\alpha}{\delta_t}\leq 2n+2\sqrt{n\kappa_t}=O(n+\sqrt{n\kappa_t})$.

To bound C_{IDC} and C_{IFC}, it suffices to note that

$$\frac{\frac{\alpha^2}{\delta_t^2}p}{\frac{L}{\delta_t}+(1-p)(\frac{\alpha}{\delta_t}+1)} \overset{(a)}{=} \frac{(\frac{\alpha}{\delta_t}+1)^2}{n(\frac{\alpha}{\delta_t}+\kappa_t)} \overset{(b)}{\leq} \frac{(2n+2\sqrt{n\kappa_t}+1)^2}{n(2n+2\sqrt{n\kappa_t}+\kappa_t)} \leq 6,$$

where (a) uses the cubic equation and (b) holds because $\frac{x+1}{x+\kappa_t}$ increases monotonically as x increases. Then,

$$\begin{aligned} C_{\text{IDC}} &\leq L^2+6L\delta_t = O\big((L+\delta_t)^2\big),\\ C_{\text{IFC}} &\leq 14L = O(L). \end{aligned}$$

D.3 Proof to Proposition 5.2

Using the interpolation condition (2) of h^{δ_t} at $(x^\star_{\delta_t}, y_k)$, we obtain

$$\begin{aligned} h^{\delta_t}(y_k) &\leq \left\langle \nabla h^{\delta_t}(y_k), y_k - x^\star_{\delta_t}\right\rangle - \frac{1}{2L}\left\|\nabla h^{\delta_t}(y_k)\right\|^2 \\ &\overset{(a)}{\leq} \frac{1-\tau_x}{\tau_x}\left\langle \nabla h^{\delta_t}(y_k), \tilde{x}_k - y_k\right\rangle + \frac{\tau_z}{\tau_x}\left\langle \nabla h^{\delta_t}(y_k), \delta_t(\tilde{x}_k - z_k) - \nabla f^{\delta_t}(\tilde{x}_k)\right\rangle \\ &\quad + \left\langle \nabla h^{\delta_t}(y_k), z_k - x^\star_{\delta_t}\right\rangle - \frac{1}{2L}\left\|\nabla h^{\delta_t}(y_k)\right\|^2 \\ &\overset{(b)}{=} \frac{1-\tau_x}{\tau_x}\left\langle \nabla h^{\delta_t}(y_k), \tilde{x}_k - y_k\right\rangle - \frac{\tau_z}{\tau_x}\left\langle \nabla h^{\delta_t}(y_k), \nabla h^{\delta_t}(\tilde{x}_k)\right\rangle \\ &\quad + \left(1-\frac{\delta_t\tau_z}{\tau_x}\right)\left\langle \nabla h^{\delta_t}(y_k), z_k - x^\star_{\delta_t}\right\rangle - \frac{1}{2L}\left\|\nabla h^{\delta_t}(y_k)\right\|^2, \end{aligned}$$

where (a) follows from the construction $y_k = \tau_x z_k + (1-\tau_x)\,\tilde{x}_k + \tau_z\left(\delta_t(\tilde{x}_k - z_k) - \nabla f^{\delta_t}(\tilde{x}_k)\right)$ and (b) uses that $\delta_t(\tilde{x}_k - z_k) - \nabla f^{\delta_t}(\tilde{x}_k) = \delta_t(x^\star_{\delta_t} - z_k) - \nabla h^{\delta_t}(\tilde{x}_k)$.

Using Lemma 1 with $\mathcal{H}_y = \mathcal{H}_k^{\delta_t}, \mathcal{G}_y = \mathcal{G}_k^{\delta_t}, z^+ = z_{k+1}, x^\star = x^\star_{\delta_t}$ and taking the expectation (note that $\mathbb{E}_{i_k}\left[\mathcal{H}_k^{\delta_t}\right] = \nabla h^{\delta_t}(y_k)$), we can conclude that

$$\begin{aligned} h^{\delta_t}(y_k) &\leq \frac{1-\tau_x}{\tau_x}\left\langle \nabla h^{\delta_t}(y_k), \tilde{x}_k - y_k\right\rangle - \frac{\tau_z}{\tau_x}\left\langle \nabla h^{\delta_t}(y_k), \nabla h^{\delta_t}(\tilde{x}_k)\right\rangle - \frac{1}{2L}\left\|\nabla h^{\delta_t}(y_k)\right\|^2 \\ &\quad + \left(1-\frac{\delta_t\tau_z}{\tau_x}\right)\frac{\alpha}{2}\left(\left\|z_k - x^\star_{\delta_t}\right\|^2 - \left(1+\frac{\delta_t}{\alpha}\right)^2\mathbb{E}_{i_k}\left[\left\|z_{k+1} - x^\star_{\delta_t}\right\|^2\right]\right) \\ &\quad + \left(1-\frac{\delta_t\tau_z}{\tau_x}\right)\frac{1}{2\alpha}\mathbb{E}_{i_k}\left[\left\|\mathcal{H}_k^{\delta_t}\right\|^2\right]. \end{aligned}$$

To bound the shifted moment, we use the interpolation condition (2) of $h_{i_k}^{\delta_t}$ at $(\tilde{x}_k, y_k)$, that is

$$\begin{aligned} \mathbb{E}_{i_k}\left[\left\|\mathcal{H}_k^{\delta_t}\right\|^2\right] &= \mathbb{E}_{i_k}\left[\left\|\nabla h_{i_k}^{\delta_t}(y_k) - \nabla h_{i_k}^{\delta_t}(\tilde{x}_k)\right\|^2\right] + 2\left\langle \nabla h^{\delta_t}(y_k), \nabla h^{\delta_t}(\tilde{x}_k)\right\rangle \\ &\quad - \left\|\nabla h^{\delta_t}(\tilde{x}_k)\right\|^2 \\ &\leq 2L\big(h^{\delta_t}(\tilde{x}_k) - h^{\delta_t}(y_k) - \left\langle \nabla h^{\delta_t}(y_k), \tilde{x}_k - y_k\right\rangle\big) \\ &\quad + 2\left\langle \nabla h^{\delta_t}(y_k), \nabla h^{\delta_t}(\tilde{x}_k)\right\rangle - \left\|\nabla h^{\delta_t}(\tilde{x}_k)\right\|^2. \end{aligned}$$

Re-arrange the terms.

$$\begin{aligned}
h^{\delta_t}(y_k) \leq\ & \left(1-\frac{\delta_t\tau_z}{\tau_x}\right)\frac{L}{\alpha}\big(h^{\delta_t}(\tilde{x}_k)-h^{\delta_t}(y_k)\big) \\
& + \left(\frac{1-\tau_x}{\tau_x}-\left(1-\frac{\delta_t\tau_z}{\tau_x}\right)\frac{L}{\alpha}\right)\left\langle \nabla h^{\delta_t}(y_k), \tilde{x}_k - y_k\right\rangle \\
& + \left(1-\frac{\delta_t\tau_z}{\tau_x}\right)\frac{\alpha}{2}\left(\left\|z_k - x^\star_{\delta_t}\right\|^2 - \left(1+\frac{\delta_t}{\alpha}\right)^2\mathbb{E}_{i_k}\left[\left\|z_{k+1}-x^\star_{\delta_t}\right\|^2\right]\right) \\
& + \left(\frac{1}{\alpha}-\frac{\delta_t\tau_z}{\alpha\tau_x}-\frac{\tau_z}{\tau_x}\right)\left\langle \nabla h^{\delta_t}(y_k), \nabla h^{\delta_t}(\tilde{x}_k)\right\rangle - \frac{1}{2L}\left\|\nabla h^{\delta_t}(y_k)\right\|^2 \\
& - \left(\frac{1}{2\alpha}-\frac{\delta_t\tau_z}{2\alpha\tau_x}\right)\left\|\nabla h^{\delta_t}(\tilde{x}_k)\right\|^2.
\end{aligned}$$

The choice of τ_z in Proposition 5.1 ensures that $\frac{1-\tau_x}{\tau_x} = \left(1-\frac{\delta_t\tau_z}{\tau_x}\right)\frac{L}{\alpha}$, which leads to

$$\begin{aligned}
h^{\delta_t}(y_k) \leq\ & (1-\tau_x)h^{\delta_t}(\tilde{x}_k) + \frac{\alpha^2(1-\tau_x)}{2L}\left(\left\|z_k - x^\star_{\delta_t}\right\|^2 - \left(1+\frac{\delta_t}{\alpha}\right)^2\mathbb{E}_{i_k}\left[\left\|z_{k+1}-x^\star_{\delta_t}\right\|^2\right]\right) \\
& + \frac{\alpha+\delta_t-(\alpha+L+\delta_t)\tau_x}{L\delta_t}\left\langle \nabla h^{\delta_t}(y_k), \nabla h^{\delta_t}(\tilde{x}_k)\right\rangle - \frac{\tau_x}{2L}\left\|\nabla h^{\delta_t}(y_k)\right\|^2 \\
& - \frac{1-\tau_x}{2L}\left\|\nabla h^{\delta_t}(\tilde{x}_k)\right\|^2.
\end{aligned} \tag{26}$$

Substitute the choice $\tau_x = \frac{\alpha+\delta_t}{\alpha+L+\delta_t}$.

$$\begin{aligned}
h^{\delta_t}(y_k) \leq\ & \frac{L}{\alpha+L+\delta_t}h^{\delta_t}(\tilde{x}_k) \\
& + \frac{\alpha^2}{2(\alpha+L+\delta_t)}\left(\left\|z_k - x^\star_{\delta_t}\right\|^2 - \left(1+\frac{\delta_t}{\alpha}\right)^2\mathbb{E}_{i_k}\left[\left\|z_{k+1}-x^\star_{\delta_t}\right\|^2\right]\right).
\end{aligned}$$

Note that by construction, $\mathbb{E}\left[h^{\delta_t}(\tilde{x}_{k+1})\right] = p\mathbb{E}\left[h^{\delta_t}(y_k)\right] + (1-p)\mathbb{E}\left[h^{\delta_t}(\tilde{x}_k)\right]$, and thus

$$\begin{aligned}
\mathbb{E}\left[h^{\delta_t}(\tilde{x}_{k+1})\right] \leq\ & \left(1-\frac{p(\alpha+\delta_t)}{\alpha+L+\delta_t}\right)\mathbb{E}\left[h^{\delta_t}(\tilde{x}_k)\right] \\
& + \frac{\alpha^2 p}{2(\alpha+L+\delta_t)}\left(\mathbb{E}\left[\left\|z_k - x^\star_{\delta_t}\right\|^2\right] - \left(1+\frac{\delta_t}{\alpha}\right)^2\mathbb{E}\left[\left\|z_{k+1}-x^\star_{\delta_t}\right\|^2\right]\right).
\end{aligned}$$

Since α is chosen as the positive root of $\left(1-\frac{p(\alpha+\delta_t)}{\alpha+L+\delta_t}\right)\left(1+\frac{\delta_t}{\alpha}\right)^2 = 1$, defining the potential function

$$T_k \triangleq \mathbb{E}\left[h^{\delta_t}(\tilde{x}_k)\right] + \frac{\alpha^2 p}{2\big(L+(1-p)(\alpha+\delta_t)\big)}\mathbb{E}\left[\left\|z_k - x^\star_{\delta_t}\right\|^2\right], \tag{27}$$

we have $T_{k+1} \leq \left(1+\frac{\delta_t}{\alpha}\right)^{-2} T_k$.

Thus, at iteration k, the following holds,

$$\begin{aligned}
\mathbb{E}\left[h^{\delta_t}(\tilde{x}_k)\right] &\leq \left(1+\frac{\delta_t}{\alpha}\right)^{-2k}\left(h^{\delta_t}(x_0) + \frac{\alpha^2 p}{2\big(L+(1-p)(\alpha+\delta_t)\big)}\left\|x_0 - x^\star_{\delta_t}\right\|^2\right) \\
&\leq \left(1+\frac{\delta_t}{\alpha}\right)^{-2k}\left(f^{\delta_t}(x_0) - f^{\delta_t}(x^\star_{\delta_t}) + \frac{\alpha^2 p}{2\big(L+(1-p)(\alpha+\delta_t)\big)}\left\|x_0 - x^\star_{\delta_t}\right\|^2\right) \\
&\overset{(\star)}{\leq} \left(1+\frac{\delta_t}{\alpha}\right)^{-2k}\left(f(x_0) - f(x^\star) + \frac{\alpha^2 p}{2\big(L+(1-p)(\alpha+\delta_t)\big)}\left\|x_0 - x^\star_{\delta_t}\right\|^2\right),
\end{aligned}$$

where $(\star)$ uses Lemma 2 *(ii)*.

Note that using the interpolation condition (2), we have

$$\begin{aligned}
\mathbb{E}\left[h^{\delta_t}(\tilde{x}_k)\right] &\geq \frac{1}{2L}\mathbb{E}\left[\left\|\nabla h^{\delta_t}(\tilde{x}_k)\right\|^2\right] \\
&= \frac{1}{2L}\mathbb{E}\left[\left\|\nabla f^{\delta_t}(\tilde{x}_k) - \delta_t(\tilde{x}_k - x^\star_{\delta_t})\right\|^2\right] \\
&= \frac{1}{2L}\mathbb{E}\left[\left\|\nabla f(\tilde{x}_k) + \delta_t(\tilde{x}_k - x_0) - \delta_t(\tilde{x}_k - x^\star_{\delta_t})\right\|^2\right] \\
&= \frac{1}{2L}\mathbb{E}\left[\left\|\nabla f(\tilde{x}_k) - \delta_t(x_0 - x^\star_{\delta_t})\right\|^2\right] \\
&\geq \frac{1}{2L}\mathbb{E}\left[\left\|\nabla f(\tilde{x}_k) - \delta_t(x_0 - x^\star_{\delta_t})\right\|\right]^2.
\end{aligned}$$

Finally, we conclude that

$$\begin{aligned}
\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|\right] &\leq \delta_t \left\|x_0 - x^\star_{\delta_t}\right\| \\
&\quad + \left(1 + \frac{\delta_t}{\alpha}\right)^{-k} \sqrt{2L\big(f(x_0) - f(x^\star)\big) + \frac{L\alpha^2 p}{L + (1-p)(\alpha + \delta_t)}\left\|x_0 - x^\star_{\delta_t}\right\|^2}.
\end{aligned} \tag{28}$$

Under IDC: Invoking Lemma 2 *(iii)* to upper bound (28), we obtain that for any $x^\star \in \mathcal{X}^\star$,

$$\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|\right] \leq \left(\delta_t + \left(1 + \frac{\delta_t}{\alpha}\right)^{-k}\sqrt{L^2 + \frac{L\alpha^2 p}{L + (1-p)(\alpha+\delta_t)}}\right)\|x_0 - x^\star\|.$$

Under IFC: Invoking Lemma 2 *(iv)* to upper bound (28), we can conclude that

$$\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|\right] \leq \left(\sqrt{2\delta_t} + \left(1 + \frac{\delta_t}{\alpha}\right)^{-k}\sqrt{2L + \frac{2L\alpha^2 p}{\big(L + (1-p)(\alpha+\delta_t)\big)\delta_t}}\right)\sqrt{f(x_0) - f(x^\star)}.$$

D.4 Proof to Theorem 5.1

(i) At outer iteration ℓ, if Algorithm 4 breaks the inner loop (Step 10) at iteration k, by construction, we have $(1 + \frac{\delta_\ell}{\alpha})^{-k}\sqrt{C_{\text{IDC}}} \leq \delta_\ell$. Then, from Proposition 5.2 *(i)*,

$$\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|\right] \leq 2\delta_\ell R_0 \overset{(\star)}{\leq} \epsilon q,$$

where $(\star)$ uses $\delta_\ell \leq \delta^\star_{\text{IDC}}$. By Markov's inequality, it holds that

$$\text{Prob}\left\{\|\nabla f(\tilde{x}_k)\| \geq \epsilon\right\} \leq \frac{\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|\right]}{\epsilon} \leq q,$$

which means that with probability at least $1 - q$, Algorithm 4 terminates at iteration k (Step 9) before reaching Step 10.

(ii) Note that the expected gradient complexity of each inner iteration is $p(n+2) + (1-p)2 = np + 2$. Then, for an inner loop that breaks at Step 10, its expected complexity is

$$\mathbb{E}\left[\#\text{grad}_t\right] \leq (np+2)\left(\frac{\alpha}{\delta_t} + 1\right)\log\frac{\sqrt{C_{\text{IDC}}}}{\delta_t}.$$

Substituting the choices in Proposition 5.1, we obtain

$$\mathbb{E}\left[\#\text{grad}_t\right] = O\left(\left(n + \sqrt{\frac{nL}{\delta_t}}\right)\log\frac{L+\delta_t}{\delta_t}\right).$$

Thus, the total expected complexity before Algorithm 4 terminates with high probability at outer iteration ℓ is at most (note that $\delta_t = \delta_0/\beta^t$)

$$\sum_{t=0}^{\ell}\mathbb{E}\left[\#\text{grad}_t\right] = O\left(\left(\ell n + \frac{1}{\sqrt{\beta}-1}\sqrt{\frac{nL\beta}{\delta_\ell}}\right)\log\frac{L+\delta_\ell}{\delta_\ell}\right).$$

Since outer iteration $\ell > 0$ is the first time $\delta_\ell \leq \delta^\star_{\text{IDC}}$, we have $\delta_\ell \leq \delta^\star_{\text{IDC}} \leq \delta_\ell \beta$. Moreover, noting that $\ell = O(\log \frac{\delta_0}{\delta_\ell})$ and $\delta_0 = L$, we can conclude that (omitting β)

$$\begin{aligned}\sum_{t=0}^{\ell} \mathbb{E}\left[\#\text{grad}_t\right] &= O\left(\left(n \log \frac{\delta_0}{\delta_\ell} + \sqrt{\frac{nL}{\delta_\ell}}\right) \log \frac{L+\delta_\ell}{\delta_\ell}\right) \\ &= O\left(\left(n \log \frac{LR_0}{\epsilon q} + \sqrt{\frac{nLR_0}{\epsilon q}}\right) \log \frac{LR_0}{\epsilon q}\right).\end{aligned}$$

D.5 Proof to Theorem 5.2

(i) At outer iteration ℓ, if Algorithm 4 breaks the inner loop (Step 11) at iteration k, by construction, we have $(1 + \frac{\delta_\ell}{\alpha})^{-k}\sqrt{C_{\text{IFC}}} \leq \sqrt{2\delta_\ell}$. Then, from Proposition 5.2 *(ii)*,

$$\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|\right] \leq \sqrt{8\delta_\ell \Delta_0} \overset{(\star)}{\leq} \epsilon q,$$

where $(\star)$ uses $\delta_\ell \leq \delta^\star_{\text{IFC}}$. By Markov's inequality, it holds that

$$\text{Prob}\left\{\|\nabla f(\tilde{x}_k)\| \geq \epsilon\right\} \leq \frac{\mathbb{E}\left[\|\nabla f(\tilde{x}_k)\|\right]}{\epsilon} \leq q,$$

which means that with probability at least $1-q$, Algorithm 4 terminates at iteration k (Step 9) before reaching Step 11.

(ii) Note that the expected gradient complexity of each inner iteration is $p(n+2)+(1-p)2 = np+2$. Then, for an inner loop that breaks at Step 11, its expected complexity is

$$\mathbb{E}\left[\#\text{grad}_t\right] \leq (np+2)\left(\frac{\alpha}{\delta_t}+1\right)\log\sqrt{\frac{C_{\text{IFC}}}{2\delta_t}}.$$

Substituting the choices in Proposition 5.1, we obtain

$$\mathbb{E}\left[\#\text{grad}_t\right] = O\left(\left(n + \sqrt{\frac{nL}{\delta_t}}\right)\log\frac{L}{\delta_t}\right).$$

Thus, the total expected complexity before Algorithm 4 terminates with high probability at outer iteration ℓ is at most (note that $\delta_t = \delta_0/\beta^t$)

$$\sum_{t=0}^{\ell} \mathbb{E}\left[\#\text{grad}_t\right] = O\left(\left(\ell n + \frac{1}{\sqrt{\beta}-1}\sqrt{\frac{nL\beta}{\delta_\ell}}\right)\log\frac{L}{\delta_\ell}\right).$$

Since outer iteration $\ell > 0$ is the first time $\delta_\ell \leq \delta^\star_{\text{IFC}}$, we have $\delta_\ell \leq \delta^\star_{\text{IFC}} \leq \delta_\ell \beta$. Moreover, noting that $\ell = O(\log \frac{\delta_0}{\delta_\ell})$ and $\delta_0 = L$, we can conclude that (omitting β)

$$\begin{aligned}\sum_{t=0}^{\ell} \mathbb{E}\left[\#\text{grad}_t\right] &= O\left(\left(n \log \frac{\delta_0}{\delta_\ell} + \sqrt{\frac{nL}{\delta_\ell}}\right) \log \frac{L}{\delta_\ell}\right) \\ &= O\left(\left(n \log \frac{\sqrt{L\Delta_0}}{\epsilon q} + \frac{\sqrt{nL\Delta_0}}{\epsilon q}\right) \log \frac{\sqrt{L\Delta_0}}{\epsilon q}\right).\end{aligned}$$

E Katyusha + L2S

By applying AdaptReg on Katyusha, Allen-Zhu [1] showed that the scheme outputs a point x_{s_1} satisfying $\mathbb{E}\left[f(x_{s_1})\right] - f(x^\star) \leq \epsilon_1$ in

$$O\left(n \log \frac{LR_0^2}{\epsilon_1} + \frac{\sqrt{nL}R_0}{\sqrt{\epsilon_1}}\right),$$

oracle calls for any $\epsilon_1 > 0$ (cf. Corollary 3.5 in [1]).

For L2S, Li et al. [37] proved that when using an n-dependent step size, its output x_a satisfies (cf. Corollary 3 in [37])

$$\mathbb{E}\left[\|\nabla f(x_a)\|\right]^2 \le \mathbb{E}\left[\|\nabla f(x_a)\|^2\right] = O\left(\frac{\sqrt{n}L\big(f(x_0) - f(x^\star)\big)}{T}\right),$$

after running T iterations.

We can combine these two rates following the ideas in [42]. Set $\epsilon_1 = O\big(\frac{T\epsilon^2}{\sqrt{n}L}\big)$ for some $\epsilon > 0$ and let the input x_0 of L2S be the output x_{s_1} of Katyusha. By chaining the above two results, we obtain the guarantee $\mathbb{E}\left[\|\nabla f(x_a)\|\right] = O(\epsilon)$ in oracle complexity

$$O\left(n + T + n\log\frac{n^{1/4}LR_0}{\sqrt{T}\epsilon} + \frac{n^{3/4}LR_0}{\sqrt{T}\epsilon}\right).$$

Minimizing the complexity by choosing $T = O\big(\frac{\sqrt{n}(LR_0)^{2/3}}{\epsilon^{2/3}}\big)$, we get the total oracle complexity

$$O\left(n\log\frac{LR_0}{\epsilon} + \frac{\sqrt{n}(LR_0)^{2/3}}{\epsilon^{2/3}}\right).$$