Accelerated Cyclic Coordinate Dual Averaging with Extrapolation for Composite Convex Optimization

Cheuk Yin Lin [1] **Chaobing Song** [1] **Jelena Diakonikolas** [1]

Abstract

Exploiting partial first-order information in a cyclic way is arguably the most natural strategy to obtain scalable first-order methods. However, despite their wide use in practice, cyclic schemes are far less understood from a theoretical perspective than their randomized counterparts. Motivated by a recent success in analyzing an extrapolated cyclic scheme for generalized variational inequalities, we propose an *Accelerated Cyclic Coordinate Dual Averaging with Extrapolation* (A-CODER) method for composite convex optimization, where the objective function can be expressed as the sum of a smooth convex function accessible via a gradient oracle and a convex, possibly nonsmooth, function accessible via a proximal oracle. We show that A-CODER attains the optimal convergence rate with improved dependence on the number of blocks compared to prior work. Furthermore, for the setting where the smooth component of the objective function is expressible in a finite sum form, we introduce a variance-reduced variant of A-CODER, VR-A-CODER, with state-of-the-art complexity guarantees. Finally, we demonstrate the effectiveness of our algorithms through numerical experiments.

1. Introduction

Block coordinate descent methods are broadly used in machine learning due to their effectiveness on large datasets brought by cheap iterations requiring only partial access to problem information (Wright, 2015; Nesterov, 2012). They are frequently applied to problems such as feature selection (Wu et al., 2008; Friedman et al., 2010; Mazumder et al., 2011), empirical risk minimization (Nesterov, 2012; Zhang & Lin, 2015; Lin et al., 2015; Allen-Zhu et al., 2016; Alacaoglu et al., 2017; Gürbüzbalaban et al., 2017; Diakonikolas & Orecchia, 2018), and in distributed computing (Liu et al., 2014; Fercoq & Richtárik, 2015; Richtárik & Takáč, 2016). In the more recent literature, coordinate updates on either the primal or the dual side in primal-dual settings have been used to attain variance-reduced guarantees in finite sum settings (Chambolle et al., 2018; Alacaoglu et al., 2017; 2020; Song et al., 2020; 2021b).

Most of the existing theoretical results for (block) coordinate-type methods have been established for algorithms that select coordinate blocks to be updated by random sampling without replacement (Nesterov, 2012; Wright, 2015; Chambolle et al., 2018; Alacaoglu et al., 2017; 2020; Song et al., 2020; 2021b; Zhang & Lin, 2015; Lin et al., 2015; Allen-Zhu et al., 2016; Diakonikolas & Orecchia, 2018). Such methods are commonly referred to as the randomized block coordinate methods (RBCMs). What makes these methods particularly appealing from the aspect of convergence analysis is that the gradient evaluated on the sampled coordinate block can be related to the full gradient, by taking the expectation over the random choice of a coordinate block.

An alternative class of block coordinate methods is the class of cyclic block coordinate methods (CBCMs), which update blocks of coordinates in a cyclic order. CBCMs are frequently used in practice due to often superior empirical performance compared to RBCMs (Beck & Tetruashvili, 2013; Chow et al., 2017; Sun & Ye, 2019) and are also part of standard software packages for high-dimensional computational statistics such as GLMNet (Friedman et al., 2010) and SparseNet (Mazumder et al., 2011). However, CBCMs have traditionally been considered much more challenging to analyze than RBCMs.

The first convergence rate analysis of CBCMs for smooth convex optimization problems, obtained by Beck & Tetruashvili (2013), relied on relating the partial coordinate blocks of the gradient to the full gradient. For this reason, the dependence of iteration complexity on the number of coordinate blocks in Beck & Tetruashvili (2013) scaled linearly and as a square root for vanilla CBCM and its accelerated variant, respectively. Such a high dependence on

[1]Department of Computer Sciences, University of Wisconsin-Madison.. Correspondence to: Cheuk Yin Lin <cylin@cs.wisc.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

the number of blocks (equal to the dimension in the coordinate case) makes the complexity guarantee of CBCMs seem worse than not only RBCMs but even full gradient methods such as gradient descent and the fast gradient method of Nesterov (1983), bringing into question their usefulness. This is further exacerbated by a result that shows that such a high gap in complexity does happen in the worst case (Sun & Ye, 2019), prompting research that would explain the gap between the theory and practice of CBCMs. However, most of the results that improved the dependence on the number of blocks only did so for structured classes of convex quadratic problems (Wright & Lee, 2020; Lee & Wright, 2019; Gürbüzbalaban et al., 2017).

On the other hand, a very recent work in Song & Diakonikolas (2021) introduced an extrapolated CBCM for variational inequalities whose complexity guarantee does not involve explicit dependence on the number of blocks. This result is enabled by a novel Lipschitz condition introduced in the same work. While the result from Song & Diakonikolas (2021) applies to convex minimization settings as a special case, the obtained convergence rates are not accelerated. Our main motivation in this work is to close this convergence gap by providing accelerated extrapolated CBCMs for convex composite minimization.

1.1. Contributions

We study the following composite convex problem

$$\min_{x \in \mathbb{R}^d} \Big\{ F(x) = f(x) + g(x) \Big\}, \tag{P}$$

where f is smooth and convex and g is proper, (possibly strongly) convex, and lower semicontinuous. This is a standard and broadly studied setting of structured nonsmooth optimization; see, e.g, Beck & Teboulle (2009); Nesterov (2007) and the follow-up work. To further make the problem amenable to optimization via block coordinate methods, we assume that g is block separable, with each component function admitting an efficiently computable prox operator (see Section 2 for a precise statement of the assumptions).

Similar to Song & Diakonikolas (2021), we define a summary Lipschitz constant L of f obtained from Lipschitz conditions of individual blocks. Our summary Lipschitz condition is similar to that of Song & Diakonikolas (2021) (although not exactly the same) and enjoys the same favorable properties as the condition introduced in that paper; see Section 2 for more details.

We introduce a new accelerated cyclic algorithm for (P) whose full gradient oracle complexity (number of full gradient passes or, equivalently, number of full cycles) is of the order $O\Big(\min \Big\{ \sqrt{\frac{L}{\gamma}} \|x_0 - x^*\|_2, \sqrt{\frac{L}{\gamma}} \log\big(\frac{L\|x_0 - x^*\|_2}{\varepsilon}\big) \Big\} \Big)$, where γ is the strong convexity parameter of g (equal to zero if g is only convex, by convention), x^* is an optimal solution to (P), and $x_0 \in \mathrm{dom}(g)$ is an arbitrary initial point. This complexity result matches the gradient oracle complexity of the fast gradient method (Nesterov, 1983), but with the traditional Lipschitz constant being replaced by the Lipschitz constant introduced in our work. In the very worst case, this constant is no higher than $\sqrt{m}$ times the traditional one, where m is the number of blocks, giving an $m^{1/4}$ improvement in the resulting complexity over the accelerated cyclic method from Beck & Tetruashvili (2013). Even in this worst case, the obtained improvement in the dependence on the number of blocks is the first such improvement for accelerated methods since the work of Beck & Tetruashvili (2013). We note, however, that for both synthetic data and real data sets and on an example problem where both Lipschitz constants are explicitly computable, our Lipschitz constant is within a small constant factor (smaller than 1.5) of the traditional one (see Figure 1, Table 1, and the related discussion in Section 2).

Some key ingredients in our analysis are the following. First, we construct an estimate of the optimality gap we want to bound, where we replace the gradient terms with a vector composed of partial, or block, extrapolated gradient terms evaluated at intermediate points within a cycle. Crucially, we show that the error introduced by doing so can be controlled and bounded via our Lipschitz condition. An auxiliary result allowing us to carry out the analysis and appropriately bound the error terms resulting from our approach is Lemma 1, which shows that our Lipschitz condition translates into inequalities of the form

$$f(y) \quad f(x) \quad \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2,$$
$$\|\nabla f(y) - \nabla f(x)\|^2 \quad 2L(f(y) - f(x) - \langle \nabla f(x), y - x \rangle),$$

similar to the standard inequalities that hold for the traditional, full-gradient, Lipschitz constant. Finally, we note that the accelerated algorithm that we introduce is novel even in the single block (i.e., full-gradient) setting, due to the employed gradient extrapolation.

We further consider the finite sum setting, where f is expressible as $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$, and where n is typically very large. We then propose a variance-reduced variant of our accelerated method, which further reduces the full gradient oracle complexity to $O\Big(\min \Big\{ \sqrt{\frac{L}{n}} \|x_0 - x^*\|_2, \sqrt{\frac{L}{n}} \log\big(\frac{L\|x_0 - x^*\|_2}{\varepsilon}\big) \Big\} \Big)$. The variance reduction that we employ is of the SVRG type (Johnson & Zhang, 2013). While following a similar approach as the basic accelerated algorithm described above, the analysis in this case turns out to be much more technical, due to the need to simultaneously handle error terms arising from variance reduction as well as the error terms arising from the cyclic updates. Through utilizing the novel smoothness properties obtained in Lemma 1 specific to convex minimization,

we are able to obtain the desired error bounds without using the additional point extrapolation step in the gradient estimator as Song & Diakonikolas (2021), but rather only with an SVRG estimator. This important change paves a path to achieving accelerated convergence rates while also simplifying the implementation of our algorithms.

Last but not least, we demonstrate the practical efficacy of our novel accelerated algorithms A-CODER and VR-A-CODER through numerical experiments, comparing against other relevant block coordinate descent methods. The use of A-CODER and VR-A-CODER achieves faster convergence in primal gap with respect to both the number of full-gradient evaluations and wall-clock time.

1.2. Further Discussion of Related Work

As discussed at the beginning of this section, cyclic block coordinate methods constitute a fundamental class of optimization methods whose convergence is not yet well understood. In the worst case, the full gradient oracle complexity of vanilla cyclic block coordinate gradient update is worse than that of vanilla gradient descent, by a factor scaling with the number of blocks m (equal to the dimension in the coordinate case) (Sun & Ye, 2019; Beck & Tetruashvili, 2013). Since the initial results providing such an upper bound (Sun & Ye, 2019), there were no improvements on the dependence on the dependence on the number of blocks in the convergence guarantees of cyclic methods until the very recent work of Song & Diakonikolas (2021), which in the worst case improves the dependence on m by a factor $\sqrt{m}$. Our work further contributes to this line of work by improving the dependence on m in accelerated methods from $\sqrt{m}$ to $m^{1=4}$ in the worst case.

In the finite-sum settings, variance reduction has been widely explored; e.g., in Johnson & Zhang (2013); Defazio et al. (2014); Allen-Zhu (2017); Reddi et al. (2016); Lei et al. (2017); Song et al. (2020); Schmidt et al. (2017) for the case of full-gradient methods and in Chen & Gu (2016); Lei & Shanbhag (2018) for randomized block coordinate methods. However, variance reduced schemes for cyclic methods are much more rare, with nonasymptotic guarantees being obtained very recently for the case of variational inequalities (Song & Diakonikolas, 2021) and nonconvex optimization (Cai et al., 2022; Xu & Yin, 2014). We are not aware of any existing variance reduced results for accelerated cyclic block coordinate methods.

1.3. Outline of the Paper

Section 2 introduces the necessary notation and background and outlines our main problem assumptions. Section 3 introduces the A-CODER algorithm and outlines the analysis. For space constraints, the full convergence analysis of A-CODER is provided in Appendix A. Section 4 presents VR-A-CODER and outlines its convergence analysis, while the full technical details are deferred to Appendix B. Finally, Section 5 provides numerical experiments for our results and concludes the paper with a discussion.

2. Notation and Preliminaries

For a positive integer K, we use $[K]$ to denote the set $\{1, 2, \ldots, K\}$. We consider the d-dimensional Euclidean space $(\mathbb{R}^d, \|\cdot\|)$, where $\|x\| = \sqrt{\langle x, x\rangle}$ denotes the Euclidean norm, $\langle \cdot, \cdot \rangle$ denotes the (standard) inner product, and d is assumed to be finite. Throughout the paper, we assume that there is a given partition of the set $\{1, 2, \ldots, d\}$ into sets S^j, $j \in \{1, \ldots, m\}$, where $|S^j| = d^j > 0$. For convenience of notation, we assume that sets S^j are comprised of consecutive elements from $\{1, 2, \ldots, d\}$, that is, $S^1 = \{1, 2, \ldots, d^1\}$, $S^2 = \{d^1 + 1, d^1 + 2, \ldots, d^1 + d^2\}, \ldots, S^m = \{\sum_{j=1}^{m-1} d^j + 1, \sum_{j=1}^{m-1} d^j + 2, \ldots, \sum_{j=1}^{m} d^j\}$. This assumption is without loss of generality, as all our results are invariant to permutations of the coordinates (though the value of the Lipschitz constant of the gradients defined in our work depends on the ordering of the coordinates; see Assumption 2). For a vector $x \in \mathbb{R}^d$, we use $x^{(j)}$ to denote its coordinate components indexed by S^j. Similarly for a gradient ∇f of a function $f : \mathbb{R}^d \to \mathbb{R}$, we use $\nabla^{(j)} f$ to denote its coordinate components indexed by S^j. We use $(\cdot)_{\downarrow j}$ to denote an operator for vectors and square matrices that *replaces the first $j-1$ elements of rows and columns with zeros*, i.e., keeping elements with indices $\geq j$ the same, otherwise zeros.

Given a proper, convex, lower semicontinuous function $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$, we use $\partial g(x)$ to denote the subdifferential set (the set of all subgradients) of g. Of particular interests to us are functions g whose proximal operator (or resolvent), defined by

$$\text{prox}_g(u) := \arg\min_{x \in \mathbb{R}^d} \left\{ \tau g(x) + \frac{1}{2}\|x - u\|^2 \right\} \quad (1)$$

is efficiently computable for all $\tau > 0$ and $u \in \mathbb{R}^d$. To unify the cases in which g are convex and strongly convex respectively, we say that g is γ-strongly convex with modulus $\gamma \geq 0$, if for all $x, y \in \mathbb{R}^d$ and $g'(x) \in \partial g(x)$,

$$g(y) \geq g(x) + \langle g'(x), y - x\rangle + \frac{\gamma}{2}\|y - x\|^2.$$

Problem definition. We consider Problem (P), under the following assumptions.

Assumption 1. *$g(x)$ is γ-strongly convex, where $\gamma \geq 0$, and block-separable over coordinate sets $\{S^j\}_{j=1}^m$: $g(x) = \sum_{j=1}^m g^j(x^{(j)})$. Each $g^j(x^{(j)})$ for $j \in [m]$ admits an efficiently computable proximal operator.*

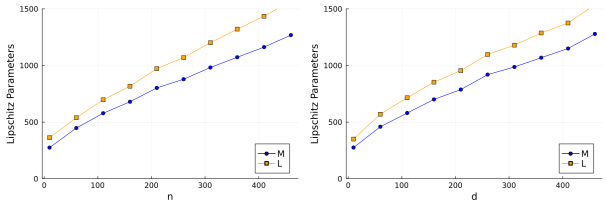

Figure 1. Comparisons of Lipschitz constants for elastic-net problems on synthetic datasets, where M denotes the commonly known Lipschitz constant and L is our new Lipschitz constant as defined in Assumption 2.

Table 1. Comparisons of Lipschitz constants for elastic-net problems on LibSVM datasets. M is the classical gradient Lipschitz constant and L is our novel smoothness constant. We use each coordinate as a block, i.e., $m = d$.

DATASET	#FEATURES	M	L
SONAR	60	12.5	15.8
COLON	2000	310.6	394.7
A9A	123	6.1	7.7
PHISHING	68	0.60	0.76
MADELON	500	1.2	1.5

Assumption 2. *There exist positive semidefinite matrices* $\{Q^1, Q^2, \ldots, Q^m\}$ *such that* $\nabla^{(j)} f(\cdot)$ *is* 1*-Lipschitz continuous w.r.t. the seminorm* $\|\cdot\|_{Q^j}$*, i.e.,* $\forall x, y \in \mathbb{R}^d$,

$$\|\nabla^{(j)} f(x) - \nabla^{(j)} f(y)\|^2 \le \|x - y\|^2_{Q^j}, \tag{2}$$

where $\|x - y\|^2_{Q^j} := (x - y)^T Q^j (x - y)$ *is the Mahalanobis (semi)norm. Moreover, we define a new Lipschitz constant* L *such that* $L^2 = 2\|Q\| < 1$ *where* $Q = \sum_{j=1}^m \left[(Q^j)_{j} + (Q^j)_{j+1}\right]$.

Observe that when f is M-smooth in a traditional sense (i.e., when f has M-Lipschitz gradients w.r.t. the Euclidean norm), Assumption 2 can be trivially satisfied using $Q^j = MI$ for all $j \in [m]$, where I is the identity matrix. Consequently, it can be argued that $L \le \sqrt{m}M$ (Song & Diakonikolas, 2021); however, we show that this bound is much tighter in practice as illustrated in Figure 1 and in Table 1. In particular, we follow the experiments in Song & Diakonikolas (2021) and show empirically that the standard Lipschitz constant M and our new Lipschitz constant L scale within the same factor for both synthetic and real data.

3. Accelerated Cyclic Algorithm

In this section, we introduce and analyze A-CODER, whose pseudocode is provided in Algorithm 1. A-CODER can be seen as a Nesterov-style accelerated variant of CODER, previously introduced for solving variational inequalities by Song & Diakonikolas (2021). A-CODER is related to other accelerated algorithms in the following sense. In the case of a single block ($m = 1$) and when gradient extrapolation is not used (i.e., when $q_k = p_k$), A-CODER reduces to a generalized variant of AGD+ (Cohen et al., 2018; Diakonikolas & Guzmán, 2021) or the method of similar triangles (Gasnikov & Nesterov, 2018). The analysis of A-CODER follows the general gap bounding argument (Diakonikolas & Orecchia, 2019; Song et al., 2021a) and it is based on three key ingredients: (i) gradient extrapolation, which enables the use of partial information about the gradients within a full epoch of cyclic updates, (ii) Lipschitz condition for the gradients based on the Mahalanobis norm as defined in Assumption 2, and (iii) upper and lower bounds on the difference between the function and its linear approximation that are compatible with the gradient Lipschitz condition that we use, as stated in Lemma 1.

Lemma 1. *Let* $f : \mathbb{R}^d \to \mathbb{R}$ *be a convex and smooth function whose gradients satisfy Assumption 2. Then,* $\forall x, y \in \mathbb{R}^d$:

$$f(y) \ge f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2,$$
$$\|\nabla f(y) - \nabla f(x)\|^2 \le 2L(f(y) - f(x) - \langle \nabla f(x), y - x \rangle).$$

We now derive the A-CODER algorithm. We define $\{a_k\}_{k \ge 1}$ and $\{A_k\}_{k \ge 1}$ to be sequences of positive numbers with $A_k = \sum_{i=1}^k a_i$, $a_0 = A_0 = 0$. Let $\{x_k\}_{k \ge 0}$ be an arbitrary sequence of points in $\mathrm{dom}(g)$. Our goal here is to bound the function value gap $f(y_k) - f(u)$ above for all $u \in \mathrm{dom}(g)$. Towards this goal, we define an estimation sequence ψ_k recursively by $\psi_0(u) = \frac{1}{2}\|u - x_0\|^2$ and

$$\psi_k(u) := \psi_{k-1}(u) + a_k(f(x_k) + \langle q_k, u - x_k \rangle + g(u))$$

for $k \ge 1$. Meanwhile, v_k and y_k are defined as $v_k := \arg\min_{u \in \mathbb{R}^d} \psi_k(u)$ and $y_k := \frac{1}{A_k}\sum_{i=1}^k a_i v_i$ respectively. We start our analysis by characterizing the gap function in the following lemma.

Lemma 2. *For any* $u \in \mathbb{R}^d$ *and any sequence of vectors* $\{q_i\}_{i \ge 1}$*, we have*

$$A_k(f(y_k) - f(u)) \tag{3}$$
$$\le \sum_{i=1}^k E_i(u) + \frac{1}{2}\|u - x_0\|^2 - \frac{1 + A_k\gamma}{2}\|u - v_k\|^2, \tag{4}$$

where

$$E_i(u) = A_i(f(y_i) - f(x_i)) - A_{i-1}(f(y_{i-1}) - f(x_i)) + a_i\langle q_i, v_i - x_i \rangle + a_i\langle \nabla f(x_i), q_i - u \rangle + \frac{1 + A_{i-1}\gamma}{2}\|v_i - v_{i-1}\|^2. \tag{5}$$

Lemma 2 applies to an arbitrary algorithm that satisfies its assumptions. From now on, we make the analysis specific

Algorithm 1 Accelerated Cyclic cOordinate Dual avEraging with extRapolation (A-CODER)

Input: $x_0 \in \mathrm{dom}(g)$, $\gamma > 0$, $L > 0$, m, $\{S^1, \ldots, S^m\}$
Initialization: $x_{-1} = x_0 = v_{-1} = v_0 = y_0$; $p_0 = \nabla f(x_0)$; $z_0 = \mathbf{0}$; $a_0 = A_0 = 0$
for $k = 1$ to K **do**
  Set $a_k > 0$ be largest value s.t. $\frac{a_k^2}{A_k} \le \frac{2(1+A_{k-1})}{5L}$ where $A_k = A_{k-1} + a_k$
  $x_k = \frac{A_{k-1}}{A_k} y_{k-1} + \frac{a_k}{A_k} v_{k-1}$
  for $j = m$ to 1 **do**
    $p_k^{(j)} = \nabla^{(j)} f(x_k^{(1)}, \ldots, x_k^{(j)}, y_k^{(j+1)}, \ldots, y_k^{(m)})$
    $q_k^{(j)} = p_k^{(j)} + \frac{a_{k-1}}{a_k}(\nabla^{(j)} f(x_{k-1}) - p_{k-1}^{(j)})$
    $z_k^{(j)} = z_{k-1}^{(j)} + a_k q_k^{(j)}$
    $v_k^{(j)} = \mathrm{prox}_{A_k g^j}(x_0^{(j)} - z_k^{(j)})$
    $y_k^{(j)} = \frac{A_{k-1}}{A_k} y_{k-1}^{(j)} + \frac{a_k}{A_k} v_k^{(j)}$
  end for
end for
return v_K, y_K

to A-CODER (Algorithm 1). In Lemma 2, $\{E_i(u)\}$ are the error terms that we need to bound above. If $\sum_{i=1}^k E_i(u) \le \frac{1+A_k}{2}\|u - v_k\|^2$, then we get the desired $1/A_k$ rate. To this end, we bound each term $E_k(u)$ in Lemma 3 by using the extrapolation direction q_k, the definition of y_k, x_k and the parameter setting of a_k.

Lemma 3. *Let $x_0 \in \mathrm{dom}(g)$ be an arbitrary initial point and consider the updates in Algorithm 1. If, for $k \ge 1$, $\frac{a_k^2}{A_k} \le \frac{2(1+A_{k-1})}{5L}$, then $\forall u$,*

$$E_k(u) \le a_k\langle \nabla f(x_k) - p_k, v_k - u\rangle + a_{k-1}\langle \nabla f(x_{k-1}) - p_{k-1}, v_{k-1} - u\rangle + \frac{1 + A_{k-1}\gamma}{10}\|v_k - v_{k-1}\|^2 + \frac{1 + A_{k-2}\gamma}{10}\|v_{k-1} - v_{k-2}\|^2.$$

We are now ready to state the main convergence result of this section.

Theorem 1. *Let $x_0 \in \mathrm{dom}(g)$ be an arbitrary initial point and consider the updates in Algorithm 1. Then, $\forall k \ge 1$ and any $u \in \mathrm{dom}(g)$:*

$$f(y_k) \le f(u) + \frac{3(1 + A_{k-1}\gamma)}{10A_k}\|u - v_k\|^2 - \frac{\|u - x_0\|^2}{2A_k}.$$

In particular, if $x^* = \arg\min_x f(x)$ exists, then

$$f(y_k) - f(x^*) \le \frac{\|x^* - x_0\|^2}{2A_k}.$$

Further, in this case we also have:

$$\|v_k - x^*\|^2 \le \frac{5}{3(1 + A_{k-1}\gamma)}\|x_0 - x^*\|^2,$$

$$\|y_k - x^*\|^2 \le \left(\frac{5}{3A_k}\sum_{i=1}^k \frac{a_i}{1 + A_{i-1}\gamma}\right)\|x_0 - x^*\|^2.$$

Finally, in all the bounds we have

$$A_k \ge \max\left\{\frac{2}{5L}\left(1 + \sqrt{\frac{2\gamma}{5L}}\right)^k, \frac{k^2}{10L}\right\}.$$

Adaptive A-CODER. The Lipschitz parameter L used in the statement of A-CODER (Algorithm 1) is usually not readily available for typical instances of convex composite minimization problems; however, as we argue in Appendix A, this parameter can be adaptively estimated using the standard backtracking line search. A variant of A-CODER implementing this adaptive estimation of L is provided in Algorithm 3. This is enabled by our analysis, which only requires the stated Lipschitz condition to hold between the successive iterates of the algorithm. Notably, unlike randomized algorithms which estimate Lipschitz constants for each of the coordinate blocks (see, e.g., Nesterov (2012)), we only need to estimate one summary Lipschitz parameter L.

4. Variance Reduced A-CODER

In this section, we assume that the problem (P) has a finite sum structure, i.e., $f(x) = \frac{1}{n}\sum_{t=1}^n f_t(x)$, where n may be very large. For this case, we can further reduce the per-iteration cost and improve the complexity results by combining the well-known SVRG-style variance reduction strategy (Johnson & Zhang, 2013) with the results from the previous section to obtain our variance reduced A-CODER (VR-A-CODER). From another perspective, VR-A-CODER can be seen as a cyclic gradient-extrapolated version of the recent VRADA algorithm for finite-sum composite convex minimization (Song et al., 2020).

For this finite-sum setting, we need to make the following stronger assumption for each $f_t(x)$.

Assumption 3. *For all $t \in [n]$, $f_t(x)$ is convex. Moreover for all $t \in [n]$, there exist positive semidefinite matrices $\{Q^1, Q^2, \ldots, Q^m\}$ such that $\nabla^{(j)} f_t(\cdot)$ is 1-Lipschitz continuous w.r.t. the norm $\|\cdot\|_{Q^j}$ i.e., $\forall x, y \in \mathbb{R}^d, t \in [n]$,*

$$\|\nabla^{(j)} f_t(x) - \nabla^{(j)} f_t(y)\|^2 \le \|x - y\|_{Q^j}^2.$$

Lemma 4. *If $f(x) = \frac{1}{n}\sum_{t=1}^n f_t(x)$ satisfies Assumption 3, then it satisfies Assumption 2 and thus Lemma 1 holds.*

Algorithm 2 Variance Reduced A-CODER (Implementable Version)

Input: $x_0 \in \mathrm{dom}(g)$, $\gamma > 0$, $L > 0$, m, $\{S^1, \ldots, S^m\}$
Initialization: $\hat{y}_0 = v_{1;0} = y_{1;0} = x_{1;1} = x_0$; $z_{1;0} = 0$
$a_0 = A_0 = 0$; $A_1 = a_1 = \frac{1}{4L}$
$z_{1;1} = \nabla f(x_0)$; $v_{1;1} = \mathrm{prox}_{a_1 g}(x_0 - z_{1;1})$
$\hat{y}_1 = y_{1;1} = v_{1;1}$
$w_{1;1;j} = (x_{1;1}^{(1)}, \ldots, x_{1;1}^{(j)}, y_{1;1}^{(j+1)}, \ldots, y_{1;1}^{(m)})$
$v_{2;0} = v_{1;1}$; $w_{2;0;j} = w_{1;1;j}$; $x_{2;0} = x_{1;1}$; $y_{2;0} = y_{1;1}$; $z_{2;0} = z_{1;1}$
for $s = 2$ to S **do**
  $a_s = \sqrt{\frac{KA_{s-1}(1+A_{s-1})}{8L}}$; $A_s = A_{s-1} + a_s$
  $a_{s;0} = a_{s-1}$; $a_{s;1} = a_{s;2} = \cdots = a_{s;K} = a_s$
  $v_{s;0} = v_{s-1;K}$; $w_{s;0;j} = w_{s-1;K;j}$; $x_{s;0} = x_{s-1;K}$; $y_{s;0} = y_{s-1;K}$; $z_{s;0} = z_{s-1;K}$
  $\tau_s = \nabla f(\hat{y}_{s-1})$
  for $k = 1$ to K **do**
    $x_{s;k} = \frac{A_{s-1}}{A_s}\hat{y}_{s-1} + \frac{a_s}{A_s} v_{s;k-1}$
    for $j = m$ to 1 **do**
      $w_{s;k;j} = (x_{s;k}^{(1)}, \ldots, x_{s;k}^{(j)}, y_{s;k}^{(j+1)}, \ldots, y_{s;k}^{(m)})$
      Choose t in $[n]$ uniformly at random
      $\tau_{s;k}^{(j)} = \nabla^{(j)} f_t(w_{s;k;j}) - \nabla^{(j)} f_t(\hat{y}_{s-1}) + \tau_s^{(j)}$
      $q_{s;k}^{(j)} = \tau_{s;k}^{(j)} + \frac{a_{s;k-1}}{a_s}(\nabla^{(j)} f_t(x_{s;k-1}) - \nabla^{(j)} f_t(w_{s;k-1;j}))$
      $z_{s;k}^{(j)} = z_{s;k-1}^{(j)} + a_s q_{s;k}^{(j)}$
      $v_{s;k}^{(j)} = \mathrm{prox}_{(A_{s-1} + \frac{a_s k}{K}) g^j}(x_0^{(j)} - z_{s;k}^{(j)}/K)$
      $y_{s;k}^{(j)} = \frac{A_{s-1}}{A_s}\hat{y}_s^{(j)} + \frac{a_s}{A_s} v_{s;k}^{(j)}$
    end for
  end for
  $\hat{y}_s = \frac{1}{K}\sum_{k=1}^{K} y_{s;k}$
end for
return $v_{S;K}, \hat{y}_S$

With this assumption, we can now derive the VR-A-CODER algorithm. Similar to Section 3, we define $\{a_s g_{s-1}$ and $\{A_s g_{s-1}$ to be sequences of positive numbers with $A_s = \sum_{i=1}^{s} a_i, a_0 = A_0 = 0$. Let $\{\hat{y}_s g_{s-0}$ be a sequence of points in $\mathrm{dom}(g)$ which will be determined by the VR-A-CODER algorithm. Our goal here is to bound the function value gap $f(\hat{y}_s) - f(u)$ above ($u \in \mathrm{dom}(g)$). To attain this, we define the estimate sequence $\{\psi_{s;k} g_{s-1;k \in [K]}$ recursively by $\psi_{1;0}(u) = \frac{K}{2}\|u - x_0\|^2$,

$$\psi_{1;1}(u) = \psi_{1;0}(u) + Ka_1(f(x_0) + \langle \nabla f(x_0), u - x_0 \rangle + g(u)), \tag{6}$$

and $\psi_{2;0} = \psi_{1;1}$; for $s = 2, 1$ k K,

$$\psi_{s;k}(u) = \psi_{s;k-1}(u) + a_s(f(x_{s;k}) + \langle q_{s;k}, u - x_{s;k} \rangle + g(u)), \tag{7}$$

and $\psi_{s+1;0} = \psi_{s;K}$. In Eqs. (6) and (7), x_0 is the initial point, $x_{s;k}$ and $y_{s;k}$ are computed as convex combinations of two points, which is commonly used in Nesterov-style acceleration, and $q_{s;k} = (q_{s;k}^{(1)}, q_{s;k}^{(2)}, \ldots, q_{s;k}^{(m)})$ is a variance reduced stochastic gradient with extrapolation, which is the main novelty in our algorithm design. Meanwhile, we define $v_{s;k}$ by $v_{s;k} := \arg\min_{u \in \mathbb{R}^d} \psi_{s;k}(u)$ and note that due to the specific choice of $q_{s;k}$, $v_{s;k}$ is updated in a cyclic (block) coordinate way. Furthermore, in VR-A-CODER, for s 2, we define $\hat{y}_s = \frac{1}{K}\sum_{k=1}^{K} y_{s;k}$.

We start our analysis by characterizing the gap function, in the following lemma, similar to Lemma 2 in Section 3, although the proof is much more technical in this case. Due to space constraints, we include the full analysis of Variance Reduced A-CODER in Appendix B.

Lemma 5. *For any $u \in \mathbb{R}^d$ and any sequence of vectors $\{q_{s;k} g_{s-2;k \in [K]}$, for all S 2, we have*

$$KA_S(f(\hat{y}_S) - f(u)) \le \frac{K}{2}\|x_0 - u\|^2 - \frac{K(1 + A_S\gamma)}{2}\|v_{S;K} - u\|^2 + \frac{K}{4}\|v_{1;1} - v_{1;0}\|^2 + \sum_{s=2}^{S}\sum_{k=1}^{K} E_{s;k}(u),$$

where

$$\begin{aligned} E_{s;k}(u) = {} & A_s(f(y_{s;k}) - f(x_{s;k})) \\ & - A_{s-1}(f(\hat{y}_{s-1}) - f(x_{s;k})) \\ & + a_s\langle \nabla f(x_{s;k}) - q_{s;k}, x_{s;k} - u \rangle \\ & + a_s\langle q_{s;k}, x_{s;k} - v_{s;k} \rangle \\ & - \frac{K(1 + A_{s-1}\gamma)}{2}\|v_{s;k} - v_{s;k-1}\|^2. \end{aligned} \tag{8}$$

In the following lemma, we bound the expected error terms $\sum_{s=2}^{S}\sum_{k=1}^{K} \mathbb{E}[E_{s;k}(u)]$ arising from the gap bound stated in the previous lemma. This bound is then finally used in Theorem 2 to obtain the claimed convergence results.

Lemma 6. *With $a_s^2 \le \frac{KA_{s-1}(1+A_{s-1})}{8L}$, $a_{s;k} = a_s$ and $A_{s;k} = A_s$ for $k \in [K]$, $a_{s;0} = a_{s-1}$ and $A_{s;0} = A_{s-1}$,*

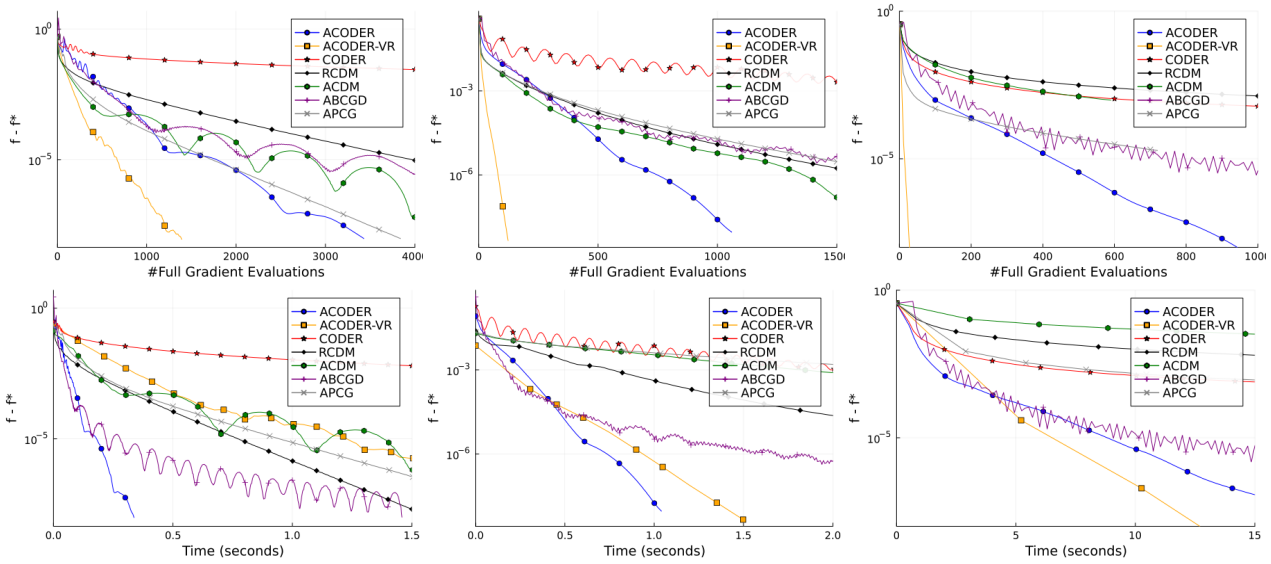

Figure 2. Performance comparisons between implemented algorithms in terms of the number of full-gradient evaluations and wall-clock time for logistic regression with ridge regularized problems. The top row contains plots against the number of full-gradient evaluations, and the bottom row contains plots against the wall-clock time. The left column is for the sonar dataset, the middle column is for the a1a dataset and the rightmost column is for the a9a dataset, all obtained from LIBSVM (Chang & Lin, 2011).

then for any fixed $u \in \mathrm{dom}(g)$ *we have*

$$\sum_{s=2}^{S}\sum_{k=1}^{K}\mathsf{E}\left[E_{s;k}(u)\right] \le \sum_{j=1}^{m} a_1\left\langle r^{(j)}f(x_{1;1}) - r^{(j)}f(w_{1;1;j}), v_{1;1}^{(j)} - u^{(j)}\right\rangle + \sum_{j=1}^{m} a_S\mathsf{E}\left[\left\langle r^{(j)}f(x_{S;K}) - r^{(j)}f(w_{S;K;j}), v_{S;K}^{(j)} - u^{(j)}\right\rangle\right] + \frac{K}{64}\|v_{1;1} - v_{1;0}\|^2 - \frac{5K(1+A_{S-1}\gamma)}{32}\mathsf{E}\left[\|v_{S;K} - v_{S;K-1}\|^2\right],$$

where $x_{1;1}, v_{1;0} \in \mathrm{dom}(g)$ *can be chosen arbitrarily and* $w_{1;1;j}$ *is defined in Algorithm 4.*

Our main result for this section is summarized in the following theorem.

Theorem 2. *Let* $x_0 \in \mathrm{dom}(g)$ *be an arbitrary initial point. Fix* $K \ge 1$ *and consider the updates in Algorithm 4. Then for* $S \ge 2$ *and* $8u \in \mathrm{dom}(g)$*, we have*

$$\mathsf{E}\left[f(\bar{y}_S) - f(u)\right] + \frac{9(1+A_{S-1}\gamma)}{64A_S}\mathsf{E}\left[\|v_{S;K} - u\|^2\right] \le \frac{5}{8A_S}\|x_0 - u\|^2.$$

In particular if $x^* = \arg\min_x f(x)$ *exists, then we have*

$$\mathsf{E}\left[f(\bar{y}_S) - f(x^*)\right] \le \frac{5}{8A_S}\|x_0 - x^*\|^2$$

and

$$\mathsf{E}\left[\|v_{S;K} - x^*\|^2\right] \le \frac{40}{9(1+A_{S-1}\gamma)}\|x_0 - x^*\|^2.$$

Finally in all the bounds above we have

$$A_S \ge \max\left\{\frac{S^2K}{64L}, \frac{1}{4L}\left(1+\sqrt{\frac{K\gamma}{8L}}\right)^{S-1}\right\}.$$

Note that in Theorem 2, we can set the number of inner iterations K to be any positive integer. However, in order to balance the computational cost between the outer loop of each epoch and the inner loops, it is optimal to set $K = \Theta(n)$ and for simplicity we can set $K = n$. Therefore, the total number of arithmetic operations required to obtain an ϵ-accurate solution $\bar{y}_S$ by applying Algorithm 2 such that $\mathsf{E}[f(\bar{y}_S) - f(x^*)] \le \epsilon$ is at most $O\left(nd\sqrt{\frac{L\|x_0 - x^*\|^2}{n}}\right)$ for the general convex case when $\gamma = 0$, and $O\left(\frac{nd\log(L\|x_0 - x^*\|^2/n\epsilon)}{\log(1+\sqrt{\gamma/L})}\right)$ for the strongly convex case when $\gamma > 0$.

Adaptive VR-A-CODER. Similar to A-CODER, VR-A-CODER can adaptively estimate the Lipschitz parameter. For completeness, we have included the adaptive version of VR-A-CODER in Algorithm 5 (Appendix B).

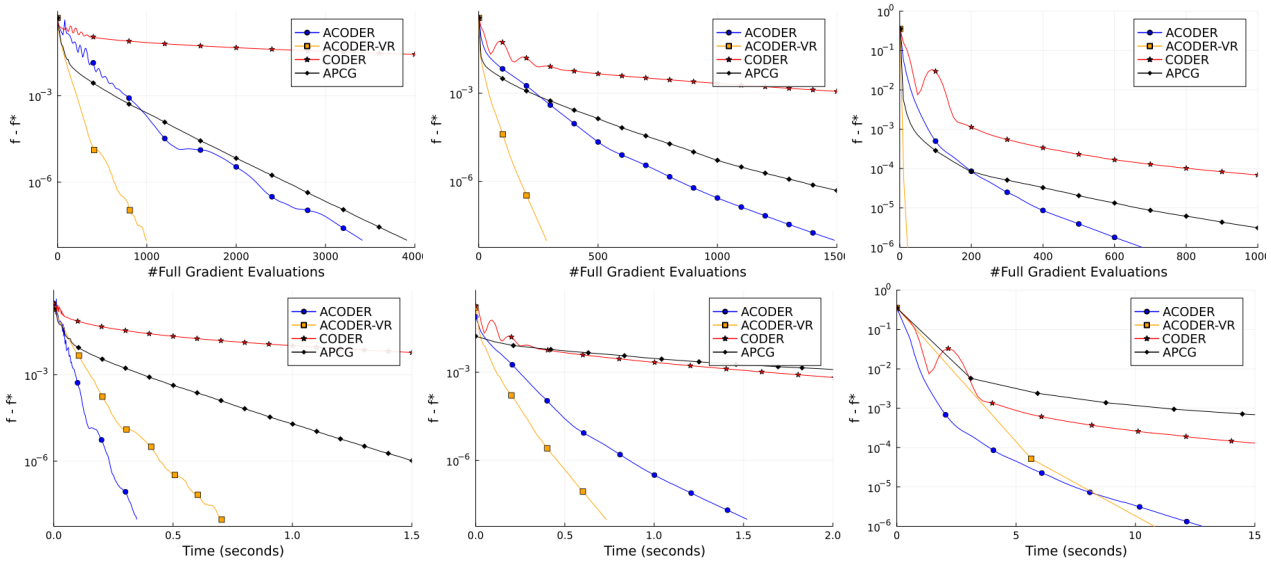

Figure 3. Performance comparisons between implemented algorithms in terms of the number of full-gradient evaluations and wall-clock time for logistic regression with elastic net regularized problems. The top row contains plots against the number of full-gradient evaluations, and the bottom row contains plots against the wall-clock time. The left column is for the sonar dataset, the middle column is for the a1a dataset and the rightmost column is for the a9a dataset, all obtained from LIBSVM (Chang & Lin, 2011).

5. Numerical Experiments and Discussion

To verify the effectiveness of our proposed algorithms, we conducted a set of numerical experiments to demonstrate that both A-CODER (Algorithm 1) and VR-A-CODER (Algorithm 2) almost completely outperform other comparable block-coordinate descent methods in terms of both iteration count and wall-clock time. In particular, we compare against a number of representative methods: CODER (Song & Diakonikolas, 2021), RCDM, ACDM (Nesterov, 2012), ABCGD (Beck & Tetruashvili, 2013) and APCG (Lin et al., 2015). For all the methods, we use the function value gap $f(x) \quad f(x^*)$ as the performance measure and we plot our results against the total number of full-gradient evaluations and against wall-clock time in seconds. We implement our experiments in Julia, a high performance programming language designed for numerical analysis and computational science, while optimizing all implementations to the best of our ability. Our code can be found at `https://github.com/ericlincc/Accelerated-CODER`.We set the block size to one in all the experiments, i.e., each block corresponds to one coordinate. We discussed in Section 4 that in theory it is optimal to choose $K = (n)$ in order to balance the computational costs of outer loop and inner loop in VR-A-CODER. We observed in our experiments that it is beneficial to choose K to be slightly smaller than n ($K \quad n/10$) to balance the computational time and the number of full-gradient evaluations.

We consider instances of ℓ_2-norm (Ridge), ℓ_1-norm (LASSO) ($\gamma = 0$) and elastic net ($\gamma > 0$) regularized logistic regression problems using three LIBSVM datasets: sonar, a1a and a9a. In the ridge regularized logistic regression problem ((Figure 2), we use $\lambda_2 = 10^{-5}$ for sonar dataset and $\lambda_2 = 10^{-4}$ for a1a and a9a datasets. In the elastic net regularized logistic regression problem (Figure 3), we use $\lambda_1 = \lambda_2 = 10^{-5}$ for sonar dataset and $\lambda_1 = \lambda_2 = 10^{-4}$ for a1a and a9a datasets. In the ℓ_1-norm regularized logistic regression problem (Figure 4), we use $\lambda_1 = 10^{-5}$ for sonar dataset and $\lambda_1 = 10^{-4}$ for a1a and a9a datasets. Figures 3 and 4 provide performance comparisons between algorithms considered in terms of the number of full-gradient evaluations and wall-clock time for the elastic net regularized logistic regression problems. We search for the best L or M for each algorithm individually at intervals of 2^i for $i \in \mathbb{Z}$, and display the best performing runs in the plots. As predicted by our theoretical results, A-CODER and VR-A-CODER exhibit accelerated convergence rates and improved dependence on the number of blocks m even in the worst case, outperforming all other algorithms. In terms of wall-clock time, due to different per-iteration cost of each algorithm in practice, we see a mildly different set of convergence behaviors. However, A-CODER and VR-A-CODER still both perform significantly better than comparable methods.

Combined with the best known theoretical convergence rates guarantee, we believe that this work provides strong supporting arguments for cyclic methods in modern machine learning applications.

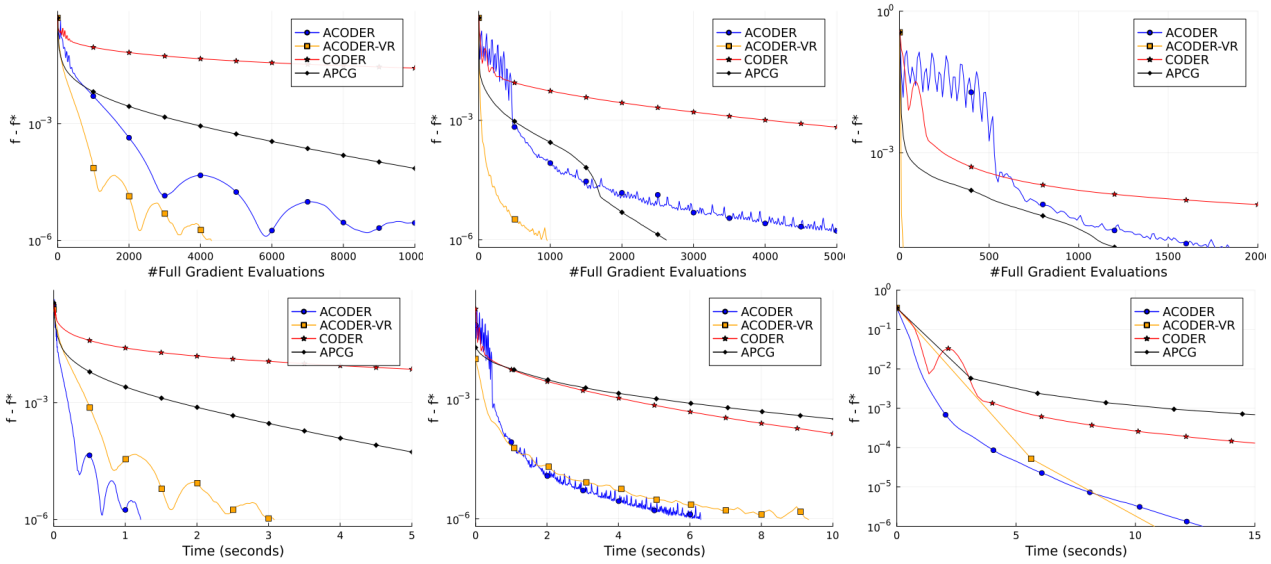

Figure 4. Performance comparisons between various algorithms in terms of number of full-gradient evaluations and wall-clock time for logistic regression with LASSO regularized problems. The top row contains plots against the number of full-gradient evaluations, and the bottom two contains plots against wall-clock time. The left column is on sonar dataset, the middle column is on a1a dataset and the rightmost column is on a9a dataset.

Acknowledgments

This research was supported in part by the NSF awards 2007757 and 2023239, and by the Office of Naval Research under contract number N00014-22-1-2348.

References

Alacaoglu, A., Dinh, Q. T., Fercoq, O., and Cevher, V. Smooth primal-dual coordinate descent algorithms for nonsmooth convex optimization. In *Proc. NIPS'17*, 2017.

Alacaoglu, A., Fercoq, O., and Cevher, V. Random extrapolation for primal-dual coordinate descent. In *Proc. ICML'20*, 2020.

Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1200–1205. ACM, 2017.

Allen-Zhu, Z., Qu, Z., Richtárik, P., and Yuan, Y. Even faster accelerated coordinate descent using non-uniform sampling. In *Proc. ICML'16*, 2016.

Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Beck, A. and Tetruashvili, L. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

Cai, X., Song, C., Wright, S. J., and Diakonikolas, J. Cyclic block coordinate descent with variance reduction for composite nonconvex optimization. *ArXiv*, abs/2212.05088, 2022.

Chambolle, A., Ehrhardt, M. J., Richtárik, P., and Schonlieb, C.-B. Stochastic primal-dual hybrid gradient algorithm

with arbitrary sampling and imaging applications. *SIAM Journal on Optimization*, 28(4):2783–2808, 2018.

Chang, C.-C. and Lin, C.-J. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Chen, J. and Gu, Q. Accelerated stochastic block coordinate gradient descent for sparsity constrained nonconvex optimization. In *Conference on Uncertainty in Artificial Intelligence*, 2016.

Chow, Y. T., Wu, T., and Yin, W. Cyclic coordinate-update algorithms for fixed-point problems: Analysis and applications. *SIAM Journal on Scientific Computing*, 39(4): A1280–A1300, 2017.

Cohen, M., Diakonikolas, J., and Orecchia, L. On acceleration with noise-corrupted gradients. In *Proc. ICML'18*, 2018.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pp. 1646–1654, 2014.

Diakonikolas, J. and Guzmán, C. Complementary composite minimization, small gradients in general norms, and applications to regression problems. *arXiv preprint arXiv:2101.11041*, 2021.

Diakonikolas, J. and Orecchia, L. Alternating randomized block coordinate descent. In *Proc. ICML'18*, 2018.

Diakonikolas, J. and Orecchia, L. The approximate duality gap technique: A unified theory of first-order methods. *SIAM Journal on Optimization*, 29(1):660–689, 2019.

Fercoq, O. and Richtárik, P. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

Friedman, J., Hastie, T., and Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

Gasnikov, A. V. and Nesterov, Y. E. Universal method for stochastic composite optimization problems. *Computational Mathematics and Mathematical Physics*, 58(1): 48–64, 2018.

Gürbüzbalaban, M., Ozdaglar, A., Parrilo, P. A., and Vanli, N. D. When cyclic coordinate descent outperforms randomized coordinate descent. In *Proc. NIPS'17*, 2017.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pp. 315–323, 2013.

Lee, C.-P. and Wright, S. J. Random permutations fix a worst case for cyclic coordinate descent. *IMA Journal of Numerical Analysis*, 39(3):1246–1275, 2019.

Lei, J. and Shanbhag, U. V. Asynchronous variance-reduced block schemes for composite non-convex stochastic optimization: block-specific steplengths and adapted batch-sizes. *Optimization Methods and Software*, 37:264 – 294, 2018.

Lei, L., Ju, C., Chen, J., and Jordan, M. I. Non-convex finite-sum optimization via SCSG methods. In *Proc. NIPS'17*, 2017.

Lin, Q., Lu, Z., and Xiao, L. An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization*, 25(4):2244–2273, 2015.

Liu, J., Wright, S., Ré, C., Bittorf, V., and Sridhar, S. An asynchronous parallel stochastic coordinate descent algorithm. In *Proc. ICML'14*, 2014.

Mazumder, R., Friedman, J. H., and Hastie, T. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495):1125–1138, 2011.

Nesterov, Y. A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2). In *Doklady AN USSR*, volume 269, pp. 543–547, 1983.

Nesterov, Y. Gradient methods for minimizing composite objective function, 2007.

Nesterov, Y. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Reddi, S. J., Hefny, A., Sra, S., Poczos, B., and Smola, A. Stochastic variance reduction for nonconvex optimization. In *Proc. ICML'16*, 2016.

Richtárik, P. and Takáč, M. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1-2):433–484, 2016.

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

Song, C. and Diakonikolas, J. Cyclic coordinate dual averaging with extrapolation for generalized variational inequalities. *arXiv preprint arXiv:2102.13244*, 2021.

Song, C., Jiang, Y., and Ma, Y. Variance reduction via accelerated dual averaging for finite-sum optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

Song, C., Jiang, Y., and Ma, Y. Unified acceleration of high-order algorithms under general hölder continuity. *SIAM Journal on Optimization*, 31(3):1797–1826, 2021a.

Song, C., Lin, C. Y., Wright, S. J., and Diakonikolas, J. Coordinate linear variance reduction for generalized linear programming. *arXiv preprint arXiv:2111.01842*, 2021b.

Sun, R. and Ye, Y. Worst-case complexity of cyclic coordinate descent: $O(n^2)$ gap with randomized version. *Mathematical Programming*, pp. 1–34, 2019.

Wright, S. and Lee, C.-p. Analyzing random permutations for cyclic coordinate descent. *Mathematics of Computation*, 89(325):2217–2248, 2020.

Wright, S. J. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

Wu, T. T., Lange, K., et al. Coordinate descent algorithms for lasso penalized regression. *Annals of Applied Statistics*, 2(1):224–244, 2008.

Xu, Y. and Yin, W. Block stochastic gradient iteration for convex and nonconvex optimization. *ArXiv*, abs/1408.2597, 2014.

Zhang, Y. and Lin, X. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *Proc. ICML'15*, 2015.

Supplementary Material for Accelerated Cyclic Coordinate Dual Averaging with Extrapolation for Composite Convex Optimization

Outline. The appendix of the paper is organized as follows:

- Section A presents the proofs related to the A-CODER algorithm in the main body of the paper, as well as the implementable and adaptive versions of A-CODER.
- Section B presents the proofs related to the A-CODER-VR algorithm in the main body of the paper. We also include the implementable and adaptive versions of A-CODER-VR in this section.

A. Omitted Proofs and Pseudocode for A-CODER

Lemma 1. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex and smooth function whose gradients satisfy Assumption 2. Then, $\forall x, y \in \mathbb{R}^d$:*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2,$$
$$\|\nabla f(y) - \nabla f(x)\|^2 \leq 2L(f(y) - f(x) - \langle \nabla f(x), y - x \rangle).$$

Proof. Let $z_j = (x^{(1)}, \ldots, x^{(j)}, y^{(j+1)}, \ldots, y^{(m)})$ and observe that $z_m = x$ and $z_0 = y$. Then we have

$$f(y) - f(x) = \sum_{j=1}^{m} (f(z_{j-1}) - f(z_j)). \tag{9}$$

As f is continuously differentiable and z_j and z_{j-1} only differ over the j^{th} block, we further have, by Taylor's theorem,

$$\begin{aligned} f(z_{j-1}) - f(z_j) &= \int_0^1 \langle \nabla f(z_j + t(z_{j-1} - z_j)), z_{j-1} - z_j \rangle \, \mathrm{d}t \\ &= \int_0^1 \left\langle \nabla^{(j)} f(z_j + t(z_{j-1} - z_j)), y^{(j)} - x^{(j)} \right\rangle \mathrm{d}t \\ &= \left\langle \nabla^{(j)} f(x), y^{(j)} - x^{(j)} \right\rangle + \int_0^1 \left\langle \nabla^{(j)} f(z_j + t(z_{j-1} - z_j)) - \nabla^{(j)} f(x), y^{(j)} - x^{(j)} \right\rangle \mathrm{d}t. \end{aligned} \tag{10}$$

Using Young's inequality, we have, for any $\alpha > 0$,

$$\begin{aligned} &\left\langle \nabla^{(j)} f(z_j + t(z_{j-1} - z_j)) - \nabla^{(j)} f(x), y^{(j)} - x^{(j)} \right\rangle \\ &\leq \frac{\alpha}{2}\|\nabla^{(j)} f(z_j + t(z_{j-1} - z_j)) - \nabla^{(j)} f(x)\|^2 + \frac{1}{2\alpha}\|y^{(j)} - x^{(j)}\|^2 \\ &\leq \frac{\alpha}{2}\|z_j + t(z_{j-1} - z_j) - x\|_{Q_j}^2 + \frac{1}{2\alpha}\|y^{(j)} - x^{(j)}\|^2 \\ &\leq \frac{\alpha}{2}\left[(1-t)\|z_j - x\|_{Q_j}^2 + t\|z_{j-1} - x\|_{Q_j}^2\right] + \frac{1}{2\alpha}\|y^{(j)} - x^{(j)}\|^2, \end{aligned}$$

where the second inequality is by our block Lipschitz assumption from Assumption 2 and the last line is by Jensen's inequality. Now observe that z_j and x agree on the first j blocks. Thus, we can write $\|z_j - x\|_{j+1} = \|y - x\|_{j+1}$ and $\|z_{j-1} - x\|_j = \|y - x\|_j$, while noting that we have $\|y - x\|_j^2 = \|y - x\|_{Q_j}^2 - \|y - x\|_{(Q_j)_{j:}}^2$ and $\|y - x\|_{j+1}^2 = \|y - x\|_{Q_j}^2 - \|y - x\|_{(Q_j)_{j+1:}}^2$. So by combining with Eq. (10) and integrating over t, we have, $\forall \alpha > 0$,

$$\begin{aligned} f(z_{j-1}) - f(z_j) &\leq \left\langle \nabla^{(j)} f(x), y^{(j)} - x^{(j)} \right\rangle + \frac{1}{2\alpha}\|y^{(j)} - x^{(j)}\|^2 \\ &\quad + \frac{\alpha}{4}\left(\|y - x\|_{(Q_j)_{j:}}^2 + \|y - x\|_{(Q_j)_{j+1:}}^2\right). \end{aligned} \tag{11}$$

Summing Eq. (11) over $j \in [m]$ and using the definition of Mahalanobis norm, we finally get

$$\begin{aligned} f(y) - f(x) &= \sum_{j=1}^{m} (f(z_{j-1}) - f(z_j)) \\ &\le \langle \nabla f(x), y - x \rangle + \frac{\alpha}{4}(y - x)^T Q(y - x) + \frac{1}{2\alpha}\|y - x\|^2 \\ &\le \langle \nabla f(x), y - x \rangle + \left(\frac{1}{2\alpha} + \frac{\alpha L^2}{8}\right)\|y - x\|^2, \end{aligned}$$

where we used Holder's inequality and the definition of L in Assumption 2. Letting $\alpha = \frac{2}{L}$ now completes the proof of the first part.

The second part of the proof is standard and is provided for completeness. Let x, y be any two points from $\mathbb{R}^d$. Define $h_x(y) = f(y) - \langle \nabla f(x), y \rangle$. Observe that $h_x(y)$ is convex (as the sum of a convex function $f(y)$ and a linear function $-\langle \nabla f(x), y \rangle$) and is minimized at $y = x$ (as for any $y \in \mathbb{R}^d$, $h_x(y) - h_x(x) = f(y) - f(x) - \langle \nabla f(x), y - x \rangle \ge 0$, by convexity of f). Observe further that for any $y, z \in \mathbb{R}^d$, we have

$$\begin{aligned} h_x(y) - h_x(z) - \langle \nabla h_x(z), y - z \rangle &= f(y) - f(z) - \langle \nabla f(z), y - z \rangle \\ &\le \frac{L}{2}\|y - z\|^2, \end{aligned}$$

where the last inequality is by the first part of the proof. The last inequality and the fact that x minimizes h_x now allow us to conclude that

$$\begin{aligned} h_x(x) &\le h_x\left(y - \frac{1}{L}\nabla h_x(y)\right) \\ &\le h_x(y) - \frac{1}{2L}\|\nabla h_x(y)\|^2. \end{aligned}$$

To complete the proof, it remains to plug the definition of $h_x(\cdot)$ into the last inequality, and rearrange. □

Lemma 2. *For any $u \in \mathbb{R}^d$ and any sequence of vectors $\{q_i\}_{i=1}$, we have*

$$A_k(f(y_k) - f(u)) \tag{3}$$

$$\le \sum_{i=1}^{k} E_i(u) + \frac{1}{2}\|u - x_0\|^2 - \frac{1 + A_k\gamma}{2}\|u - v_k\|^2, \tag{4}$$

where

$$\begin{aligned} E_i(u) &= A_i(f(y_i) - f(x_i)) - A_{i-1}(f(y_{i-1}) - f(x_i)) \\ &\quad - a_i\langle q_i, v_i - x_i \rangle + a_i\langle \nabla f(x_i), q_i - u \rangle \\ &\quad - \frac{1 + A_{i-1}\gamma}{2}\|v_i - v_{i-1}\|^2. \end{aligned} \tag{5}$$

Proof. As $y_k = \frac{1}{A_k}\sum_{i=1}^{k} a_i v_i$ and g is convex, we have $g(y_k) \le \frac{1}{A_k}\sum_{i=1}^{k} a_i g(v_i)$ and thus,

$$\begin{aligned} A_k f(y_k) &\le A_k f(y_k) + \sum_{i=1}^{k} a_i g(v_i) \\ &= \sum_{i=1}^{k} (A_i f(y_i) - A_{i-1} f(y_{i-1})) + \sum_{i=1}^{k} a_i g(v_i), \end{aligned} \tag{12}$$

where the equality is by $A_0 = 0$. Then, as f is convex and $f = f + g$, we have, $\forall u$,

$$\begin{aligned} A_k f(u) &= \sum_{i=1}^{k} a_i f(u) \quad \sum_{i=1}^{k} a_i (f(x_i) + \langle \nabla f(x_i), u \quad x_i \rangle + g(u)) \\ &= \sum_{i=1}^{k} a_i (f(x_i) + \langle q_i, u \quad x_i \rangle + g(u)) + \sum_{i=1}^{k} a_i \langle \nabla f(x_i) \quad q_i, u \quad x_i \rangle \\ &= \psi_k(u) \quad \psi_0(u) + \sum_{i=1}^{k} a_i \langle \nabla f(x_i) \quad q_i, u \quad x_i \rangle \\ &\quad \psi_k(v_k) + \frac{1 + A_k \gamma}{2} \|u \quad v_k\|^2 \quad \frac{1}{2} \|u \quad x_0\|^2 \\ &\quad + \sum_{i=1}^{k} a_i \langle \nabla f(x_i) \quad q_i, u \quad x_i \rangle , \end{aligned}$$

where the first inequality is by the convexity of f, the third equality is by the recursive definition of $\psi_k(u)$, and the last inequality is by the $(1 + A_k\gamma)$-strong convexity of $\psi_k(u)$, $v_k = \arg\min_u \psi_k(u)$ which implies $\psi_k(u) \quad \psi_k(v_k) + \frac{1+A_k}{2} \|u \quad v_k\|^2$, and the definition of $\psi_0(u)$.

Then as $\psi_0(v_0) = 0$, using the recursive definition of ψ_k, we have

$$\begin{aligned} \psi_k(v_k) &= \sum_{i=1}^{k} (\psi_i(v_i) \quad \psi_{i \quad 1}(v_{i \quad 1})) \\ &= \sum_{i=1}^{k} \Big((\psi_{i \quad 1}(v_i) \quad \psi_{i \quad 1}(v_{i \quad 1})) + a_i (f(x_i) + \langle q_i, v_i \quad x_i \rangle + g(v_i)) \Big) \\ &\quad \sum_{i=1}^{k} \Big(\frac{1 + A_{i \quad 1}\gamma}{2} \|v_i \quad v_{i \quad 1}\|^2 + a_i (f(x_i) + \langle q_i, v_i \quad x_i \rangle + g(v_i)) \Big), \end{aligned} \tag{13}$$

where the last inequality is by the $(1 + A_{i \quad 1}\gamma)$-strong convexity of $\psi_{i \quad 1}$ and the optimality of $v_{i \quad 1}$. Combining Eqs. (12) and (13), we have

$$A_k (f(y_k) \quad f(u)) \quad \sum_{i=1}^{k} E_i(u) \quad \frac{1 + A_k \gamma}{2} \|u \quad v_k\|^2 + \frac{1}{2} \|u \quad x_0\|^2, \tag{14}$$

where $E_i(u)$ is defined in (5). □

Lemma 3. *Let $x_0 \ 2 \ \mathrm{dom}(g)$ be an arbitrary initial point and consider the updates in Algorithm 1. If, for $k \quad 1$, $\frac{a_k^2}{A_k} \quad \frac{2(1+A_{k \quad 1})}{5L}$, then $\forall u$,*

$$\begin{aligned} E_k(u) \quad & a_k \langle \nabla f(x_k) \quad p_k, v_k \quad u \rangle \\ & a_{k \quad 1} \langle \nabla f(x_{k \quad 1}) \quad p_{k \quad 1}, v_{k \quad 1} \quad u \rangle \\ & \frac{1 + A_{k \quad 1}\gamma}{10} \|v_k \quad v_{k \quad 1}\|^2 \\ & + \frac{1 + A_{k \quad 2}\gamma}{10} \|v_{k \quad 1} \quad v_{k \quad 2}\|^2. \end{aligned}$$

Proof. By the convexity of f, we have

$$f(y_{k \quad 1}) \quad f(x_k) \quad \langle \nabla f(x_k), y_{k \quad 1} \quad x_k \rangle .$$

Then by applying Lemma 1, we have

$$A_k(f(y_k) \quad f(x_k)) \quad A_{k\ 1}(f(y_{k\ 1}) \quad f(x_k)) \quad \langle r f(x_k), A_k y_k \quad A_{k\ 1} y_{k\ 1} \quad a_k x_k i + \frac{A_k L}{2} k y_k \quad x_k k^2$$
$$= a_k \langle r f(x_k), v_k \quad x_k i + \frac{a_k{}^2 L}{2A_k} k v_k \quad v_{k\ 1} k^2, \tag{15}$$

where we used the definitions of y_k and x_k from Algorithm 1 in the last equality. Combining Eq. (5) (with $i = k$) in Lemma 2 and Eq. (15), we have

$$E_k(u) \quad \Big(\frac{a_k{}^2 L}{2A_k} \quad \frac{1 + A_{k\ 1}\gamma}{2}\Big) k v_k \quad v_{k\ 1} k^2 + a_k \langle r f(x_k) \quad q_k, v_k \quad u i . \tag{16}$$

Thus by rewriting the second term as the sum of inner products over the m blocks and by using the definition of $q_k^{(j)}$ in Algorithm 1, we have

$$a_k \langle r f(x_k) \quad q_k, v_k \quad u i = a_k \sum_{j=1}^{m} \Big\langle r^{(j)} f(x_k) \quad q_k^{(j)}, v_k^{(j)} \quad u^{(j)} \Big\rangle$$
$$= \sum_{j=1}^{m} \Big[a_k \Big\langle r^{(j)} f(x_k) \quad p_k^{(j)}, v_k^{(j)} \quad u^{(j)} \Big\rangle \quad a_{k\ 1} \Big\langle r^{(j)} f(x_{k\ 1}) \quad p_{k\ 1}^{(j)}, v_{k\ 1}^{(j)} \quad u^{(j)} \Big\rangle \Big] \tag{17}$$
$$+ a_{k\ 1} \sum_{j=1}^{m} \Big\langle r^{(j)} f(x_{k\ 1}) \quad p_{k\ 1}^{(j)}, v_{k\ 1}^{(j)} \quad v_k^{(j)} \Big\rangle .$$

Notice that the first two inner product terms in the first line of Eq. (17) telescope when summed over k, therefore it remains to bound $a_{k\ 1} \sum_{j=1}^{m} \Big\langle r^{(j)} f(x_{k\ 1}) \quad p_k^{(j)}, v_k^{(j)} \quad v_{k\ 1}^{(j)} \Big\rangle$. In particular we let $w_{k;j} = (x_k^1, \ldots, x_k^j, y_k^{j+1}, \ldots, y_k^m)$ so that $p_k^{(j)} = r^{(j)} f(w_{k;j})$, then we have

$$\Big\langle r^{(j)} f(x_{k\ 1}) \quad p_{k\ 1}^{(j)}, v_{k\ 1}^{(j)} \quad v_k^{(j)} \Big\rangle = \Big\langle r^{(j)} f(x_{k\ 1}) \quad r^{(j)} f(w_{k\ 1;j}), v_{k\ 1}^{(j)} \quad v_k^{(j)} \Big\rangle$$
$$\frac{\alpha}{2} \Big\| r^{(j)} f(x_{k\ 1}) \quad r^{(j)} f(w_{k\ 1;j}) \Big\|^2 + \frac{1}{2\alpha} \Big\| v_{k\ 1}^{(j)} \quad v_k^{(j)} \Big\|^2$$
$$\frac{\alpha}{2} k x_{k\ 1} \quad w_{k\ 1;j} k_{Q^j}^2 + \frac{1}{2\alpha} \Big\| v_{k\ 1}^{(j)} \quad v_k^{(j)} \Big\|^2 \tag{18}$$

where the first inequality holds for any $\alpha > 0$ by Young's inequality and the second inequality is by Assumption 2. Notice that $x_{k\ 1}$ and $w_{k\ 1;j}$ agree on the first j blocks, so similar to the proof of Lemma 1 we can write $x_{k\ 1} \quad w_{k\ 1;j} = (y_{k\ 1} \quad x_{k\ 1})_{\ j+1}$. Therefore by applying similar arguments as Lemma 1, we get

$$\sum_{j=1}^{m} k x_{k\ 1} \quad w_{k\ 1;j} k_{Q^j}^2 = \sum_{j=1}^{m} k y_{k\ 1} \quad x_{k\ 1} k_{(Q^j)_{\ j+1}}^2$$
$$\sum_{j=1}^{m} k y_{k\ 1} \quad x_{k\ 1} k_{(Q^j)_{\ j+1}}^2 + \sum_{j=1}^{m} k y_{k\ 1} \quad x_{k\ 1} k_{(Q^j)_{\ j}}^2$$
$$= k y_{k\ 1} \quad x_{k\ 1} k_Q^2$$
$$\frac{a_{k\ 1}^2 L^2}{2A_{k\ 1}^2} k v_{k\ 1} \quad v_{k\ 2} k^2 \tag{19}$$

where we used the non-negativity of Mahalanobis norm w.r.t. semi-positive definite matrix in the first inequality and the definition of x_k, y_k and L in the last inequality. Lastly, by combining Eqs. (16), (17), (18) and (19), we have

$$E_k(u) \quad a_k \hbar r f(x_k) \quad p_k, v_k \quad u i \quad a_{k\ 1} \hbar r f(x_{k\ 1}) \quad p_{k\ 1}, v_{k\ 1} \quad u i + \left(\frac{a_k^2 L}{2A_k} \quad \frac{1 + A_{k\ 1}\gamma}{2} + \frac{a_{k\ 1}}{2\alpha}\right) k v_k \quad v_{k\ 1} k^2 + \left(\frac{\alpha a_{k\ 1}^3 L^2}{4A_{k\ 1}^2}\right) k v_{k\ 1} \quad v_{k\ 2} k^2 .$$

It remains to choose $\alpha = \frac{A_{k\ 1}}{a_{k\ 1} L}$ and some sequence $f a_i g_i^k$ such that $\frac{a_k^2}{A_k} \quad \frac{2(1+A_{k\ 1})}{5L}$. □

Theorem 1. *Let x_0 2 dom(g) be an arbitrary initial point and consider the updates in Algorithm 1. Then, 8k 1 and any u 2 dom(g):*

$$f(y_k) \quad f(u) + \frac{3(1 + A_{k\ 1}\gamma)}{10A_k} k u \quad v_k k^2 \quad \frac{k u \quad x_0 k^2}{2A_k}.$$

In particular, if x = arg min$_x$ $f(x)$ exists, then

$$f(y_k) \quad f(x) \quad \frac{k x \quad x_0 k^2}{2A_k}.$$

Further, in this case we also have:

$$k v_k \quad x \ k^2 \quad \frac{5}{3(1 + A_{k\ 1}\gamma)} k x_0 \quad x \ k^2,$$

$$k y_k \quad x \ k^2 \quad \left(\frac{5}{3A_k} \sum_{i=1}^{k} \frac{a_i}{1 + A_{i\ 1}\gamma}\right) k x_0 \quad x \ k^2.$$

Finally, in all the bounds we have

$$A_k \quad \max\left\{\frac{2}{5L}\left(1 + \sqrt{\frac{2\gamma}{5L}}\right)^k, \ \frac{k^2}{10L}\right\}.$$

Proof. By Lemma 3, and using the fact $A_0 = a_0 = 0$ and $v_0 = v_{\ 1}$, we have

$$\sum_{i=1}^{k} E_i(u) \quad \frac{1 + A_{k\ 1}\gamma}{10} k v_k \quad v_{k\ 1} k^2 + a_k \hbar r f(x_k) \quad p_k, v_k \quad u i . \tag{20}$$

Same as in the proof of Lemma 3, we can bound $a_k \hbar r f(x_k) \quad p_k, v_k \quad u i$ using Young's inequality and the definition of smoothness for f. In particular, for any $\alpha > 0$,

$$a_k \hbar r f(x_k) \quad p_k, v_k \quad u i \quad a_k\left(\frac{\alpha L^2}{4} k y_k \quad x_k k^2 + \frac{1}{2\alpha} k u \quad v_k k^2\right) = a_k\left(\frac{\alpha L^2 a_k{}^2}{4A_k{}^2} k v_k \quad v_{k\ 1} k^2 + \frac{1}{2\alpha} k u \quad v_k k^2\right).$$

Choosing $\alpha = \frac{A_k}{a_k L}$ and using $\frac{a_k{}^2}{A_k} \quad \frac{2(1+A_{k\ 1})}{5L}$, we get

$$a_k \hbar r f(x_k) \quad p_k, v_k \quad u i \quad \frac{(1 + A_{k\ 1}\gamma)}{10} k v_k \quad v_{k\ 1} k^2 + \frac{(1 + A_{k\ 1}\gamma)}{5} k u \quad v_k k^2. \tag{21}$$

Then combining Lemma 2, Eq. (21) and Eq. (20) with the fact $A_{k\ 1} \quad A_k$, we have

$$(f(y_k) \quad f(u)) + \frac{3(1 + A_{k\ 1}\gamma)}{10A_k} k u \quad v_k k^2 \quad \frac{1}{2A_k} k u \quad x_0 k^2. \tag{22}$$

Assume now that x = arg min$_x$ $f(x)$ exists. As $f(y_k) \quad f(x) \quad 0$, Eq. (22) implies

$$k v_k \quad x \ k^2 \quad \frac{5}{3(1 + A_{k\ 1}\gamma)} k x_0 \quad x \ k^2. \tag{23}$$

Algorithm 3 Adaptive Accelerated Cyclic cOordinate Dual avEraging with extRapolation (Ada-A-CODER)

Input: $x_0 \in \mathrm{dom}(g), \gamma \geq 0, L_0 > 0, m, \{S^1, \ldots, S^m\}$
Initialization: $x_{-1} = x_0 = v_{-1} = v_0 = y_0, p_0 = \nabla f(x_0), z_0 = 0, a_0 = A_0 = 0$
for $k = 1$ to K **do**
  $L_k = L_{k-1}/2$
  repeat
    $L_k = 2L_k$
    Set $a_k > 0$ be largest value s.t. $\frac{a_k^2}{A_k} \leq \frac{2(1+A_{k-1})}{5L_k}$ where $A_k = A_{k-1} + a_k$
    $x_k = \frac{A_{k-1}}{A_k} y_{k-1} + \frac{a_k}{A_k} v_{k-1}$
    for $j = m$ to 1 **do**
      $p_k^{(j)} = \nabla^{(j)} f(x_k^{(1)}, \ldots, x_k^{(j)}, y_k^{(j+1)}, \ldots, y_k^{(m)})$
      $q_k^{(j)} = p_k^{(j)} + \frac{a_{k-1}}{a_k}(\nabla^{(j)} f(x_{k-1}) - p_{k-1}^{(j)})$
      $z_k^{(j)} = z_{k-1}^{(j)} + a_k q_k^{(j)}$
      $v_k^{(j)} = \mathrm{prox}_{A_k g^j}(x_0^{(j)} - z_k^{(j)})$
      $y_k^{(j)} = \frac{A_{k-1}}{A_k} y_{k-1}^{(j)} + \frac{a_k}{A_k} v_k^{(j)}$
    end for
  until $f(y_k) \leq f(x_k) + \langle \nabla f(x_k), y_k - x_k \rangle + \frac{L_k}{2}\|y_k - x_k\|^2$
end for
return v_K, y_K

Using Jensen's inequality, as $y_k = \frac{1}{A_k}\sum_{i=1}^k a_i v_i$, we also have from Eq. (23)

$$\|y_k - x^*\|^2 \leq \left(\frac{5}{3A_k}\sum_{i=1}^k \frac{a_i}{1 + A_{i-1}\gamma}\right)\|x_0 - x^*\|^2.$$

Finally, recall once again that $\{a_k g_{k-1}$ is chosen so that $\frac{a_k^2}{A_k} = \frac{2(1+A_{k-1})}{5L}$. When $\gamma = 0$, this leads to the standard $A_k \geq \frac{k^2}{10L}$ growth of accelerated algorithms by choosing $a_k = \frac{k}{5L}$ for $k \geq 1$. When $\gamma > 0$, we have $\frac{a_k}{A_{k-1}} > \sqrt{\frac{2}{5L}}$, and it remains to use that $A_k = \frac{A_k}{A_{k-1}} \cdots \frac{A_2}{A_1} A_1 = A_1\left(1 + \sqrt{\frac{2}{5L}}\right)^{k-1}$ where $a_1 = A_1 = \frac{2}{5L}$ using the choice of a_k in Algorithm 1 and $A_0 = a_0 = 0$, completing the proof. □

A.1. (Lipschitz) Parameter-Free A-CODER

Similar to CODER, it is possible to adaptively estimate the Lipschitz parameter L_k for A-CODER. Note that in the case of A-CODER, all that is needed for the analysis from Section 3 to apply is that the quadratic bound from Lemma 1 holds between x_k and y_k. A variant of A-CODER that implements this adaptive estimation is provided in Algorithm 3.

B. Omitted Proofs and Pseudocode for ACODER-VR

Lemma 4. *If $f(x) = \frac{1}{n}\sum_{t=1}^n f_t(x)$ satisfies Assumption 3, then it satisfies Assumption 2 and thus Lemma 1 holds.*

Proof. By using Jensen's inequality and Assumption 3, we have

$$\left\|\nabla^{(j)} f(x) - \nabla^{(j)} f(y)\right\|^2 \leq \frac{1}{n}\sum_{t=1}^n \left\|\nabla^{(j)} f_t(x) - \nabla^{(j)} f_t(y)\right\|^2 \leq \|x - y\|_{Q^j}^2.$$

□

Lemma 5. *For any $u \in \mathbb{R}^d$ and any sequence of vectors $\{q_{s;k}\}_{s=2;k\in[K]}$, for all $S \geq 2$, we have*

$$K A_S(f(\bar{y}_S) - f(u))$$

Algorithm 4 Variance Reduced A-CODER (Analysis Version)

Input: $x_0 \in \mathrm{dom}(g), \gamma \geq 0, L > 0, m, \{S^1, \ldots, S^m\}$
Initialization: $\hat{y}_0 = v_{1;0} = y_{1;0} = x_{1;1} = x_0$
$a_0 = A_0 = 0$; $A_1 = a_1 = \frac{1}{4L}$
$\psi_{1;0}(\cdot) = \frac{K}{2}\|\cdot - x_0\|^2$
$v_{1;1} = \arg\min_v \{\psi_{1;1}(v) := \psi_{1;0}(v) + Ka_1\langle f(x_0) + \hat{\nabla} f(x_0), v - x_0\rangle + a_1 g(v)\}$
$w_{1;1;j} = (x_{1;1}^{(1)}, \ldots, x_{1;1}^{(j)}, y^{(j+1)}_{1;1}, \ldots, y^{(m)}_{1;1})$
$\hat{y}_1 = v_{2;0} = y_{1;1} = v_{1;1}$; $w_{2;0;j} = w_{1;1;j}$; $\psi_{2;0} = \psi_{1;1}$
for $s = 2$ to S **do**
  Set $a_s > 0$ s.t. $a_s^2 = \frac{KA_{s-1}(1+A_{s-1})}{8L}$; $A_s = A_{s-1} + a_s$
  $a_{s;0} = a_{s-1}$; $a_{s;1} = a_{s;2} = \cdots = a_{s;K} = a_s$
  $x_{s;0} = x_{s-1;K}$; $y_{s;0} = x_{s-1;K}$; $w_{s;0;j} = w_{s-1;K;j}$; $v_{s;0} = v_{s-1;K}$; $\psi_{s;0} = \psi_{s-1;K}$
  $\hat{\nabla}_s = \hat{\nabla} f(\hat{y}_{s-1})$
  for $k = 1$ to K **do**
    $x_{s;k} = \frac{A_{s-1}}{A_s}\hat{y}_{s-1} + \frac{a_s}{A_s} v_{s;k-1}$
    for $j = m$ to 1 **do**
      $w_{s;k;j} = (x_{s;k}^{(1)}, \ldots, x_{s;k}^{(j)}, y_{s;k}^{(j+1)}, \ldots, y_{s;k}^{(m)})$
      Choose t in $[n]$ uniformly at random
      $\hat{\nabla}_{s;k}^{(j)} = \nabla^{(j)} f_t(w_{s;k;j}) - \nabla^{(j)} f_t(\hat{y}_{s-1}) + \hat{\nabla}_s^{(j)}$
      $q_{s;k}^{(j)} = \hat{\nabla}_{s;k}^{(j)} + \frac{a_{s;k-1}}{a_s}(\nabla^{(j)} f_t(x_{s;k-1}) - \nabla^{(j)} f_t(w_{s;k-1;j}))$
      $v_{s;k}^{(j)} = \arg\min_{v^{(j)} \in \mathbb{R}^{d^j}} \{\psi_{s;k}^j(v^{(j)}) := \psi_{s;k-1}^j(v^j) + a_s(\frac{1}{m}\langle f(x_{s;k}) + \hat{\nabla}_{s;k}^{(j)}, v^{(j)} - y^{(j)}_{s;k-1}\rangle + g^j(v^{(j)}))\}$
      $y_{s;k}^{(j)} = \frac{A_{s-1}}{A_s}\hat{y}_s^{(j)} + \frac{a_s}{A_s} v_{s;k}^{(j)}$
    end for
  end for
  $\hat{y}_s = \frac{1}{K}\sum_{k=1}^K y_{s;k}$
end for
return $v_{S;L}, \hat{y}_S$

$$\frac{K}{2}\|x_0 - u\|^2 - \frac{K(1 + A_S\gamma)}{2}\|v_{S;K} - u\|^2 - \frac{K}{4}\|v_{1;1} - v_{1;0}\|^2 + \sum_{s=2}^{S}\sum_{k=1}^{K} E_{s;k}(u),$$

where

$$\begin{aligned} E_{s;k}(u) = {} & A_s(f(y_{s;k}) - f(x_{s;k})) - A_{s-1}(f(\hat{y}_{s-1}) - f(x_{s;k})) + a_s\langle \hat{\nabla} f(x_{s;k}) - q_{s;k}, x_{s;k} - u\rangle \\ & + a_s\langle q_{s;k}, x_{s;k} - v_{s;k}\rangle - \frac{K(1 + A_{s-1}\gamma)}{2}\|v_{s;k} - v_{s;k-1}\|^2. \end{aligned} \tag{8}$$

Proof. As f is convex and $f = f + g$, we have: $\forall u$,

$$\begin{aligned} KA_S f(u) = {} & \sum_{s=1}^{S}\sum_{k=1}^{K} a_s f(u) \\ & Ka_1(f(x_{1;1}) + \langle \hat{\nabla} f(x_{1;1}), u - x_{1;1}\rangle + g(u)) \\ & + \sum_{s=2}^{S}\sum_{k=1}^{K} a_s(f(x_{s;k}) + \langle \hat{\nabla} f(x_{s;k}), u - x_{s;k}\rangle + g(u)) \\ = {} & \psi_{S;K}(u) - \psi_{1;0}(u) + \sum_{s=2}^{S}\sum_{k=1}^{K} a_s\langle \hat{\nabla} f(x_{s;k}) - q_{s;k}, u - x_{s;k}\rangle \end{aligned}$$

$$\begin{aligned}
&\psi_{s;K}(v_{s;K}) + \frac{K(1+A_s\gamma)}{2}\|u - v_{s;K}\|^2 \quad \frac{K}{2}\|x_0 - u\|^2 \\
&+ \sum_{s=2}^{S}\sum_{k=1}^{K} a_s\langle \nabla f(x_{s;k}) \quad q_{s;k}, u \quad x_{s;k}\rangle ,
\end{aligned} \tag{24}$$

where the first inequality is by the convexity of f, the second equality is by the recursive definition of $\psi_{s;K}(u)$, the second inequality is by the $K(1 + A_s\gamma)$-strong convexity of $\psi_{s;K}(u)$ and $v_{s;K} = \arg\min_u \psi_{s;K}(u)$ leading to $\psi_{s;K}(u)$ $\psi_{s;K}(v_{s;K}) + \frac{K(1+A_s)}{2}\|u - v_{s;K}\|^2$.

Then using our recursive definition of the estimate sequences again, we have

$$\begin{aligned}
&\psi_{s;K}(v_{s;K}) \\
&= \psi_{1;1}(v_{1;1}) + \sum_{s=2}^{S}\sum_{k=1}^{K}(\psi_{s;k}(v_{s;k}) \quad \psi_{s;k-1}(v_{s;k-1})) \\
&= \psi_{1;1}(v_{1;1}) + \sum_{s=2}^{S}\sum_{k=1}^{K}(\psi_{s;k-1}(v_{s;k}) \quad \psi_{s;k-1}(v_{s;k-1})) \\
&\quad + \sum_{s=2}^{S}\sum_{k=1}^{K} a_s(f(x_{s;k}) + \langle q_{s;k}, v_{s;k} \quad x_{s;k}\rangle + g(v_{s;k})) \\
&\quad \frac{K}{2}\|v_{1;1} \quad v_{1;0}\|^2 + Ka_1(f(x_{1;1}) + \langle \nabla f(x_{1;1}), v_{1;1} \quad x_{1;1}\rangle + g(v_{1;1})) \\
&\quad + \sum_{s=2}^{S}\sum_{k=1}^{K}\frac{K(1+A_{s-1}\gamma)}{2}\|v_{s;k} \quad v_{s;k-1}\|^2 \\
&\quad + \sum_{s=2}^{S}\sum_{k=1}^{K} a_s(f(x_{s;k}) + \langle q_{s;k}, v_{s;k} \quad x_{s;k}\rangle + g(v_{s;k})),
\end{aligned} \tag{25}$$

where the first equality is by $\psi_{s+1;0} = \psi_{s;K}$ and $v_{s+1;0} = v_{s;K}$, the second equality is by the definition of $\psi_{s;k}$, the last inequality is by the definition of $\psi_{1;1}(v_{1;1})$ and the $K(1 + A_{s-1}\gamma)$-strong convexity of $\psi_{s;k-1}(s \quad 2, k \in [K])$. Then by Lemmas 4 and 1, we have

$$\begin{aligned}
f(v_{1;1}) \quad & f(x_{1;1}) + \langle \nabla f(x_{1;1}), v_{1;1} \quad x_{1;1}\rangle + \frac{L}{2}\|v_{1;1} \quad x_{1;1}\|^2 \\
& f(x_{1;1}) + \langle \nabla f(x_{1;1}), v_{1;1} \quad x_{1;1}\rangle + \frac{1}{4a_1}\|v_{1;1} \quad v_{1;0}\|^2,
\end{aligned} \tag{26}$$

where the last inequality is by $a_1 \quad \frac{1}{4L}$ and $v_{1;0} = x_{1;1}$.

Using $\nabla_s = \frac{1}{K}\sum_{k=1}^{K}\nabla_{s;k} = \frac{A_{s-1}}{A_s}\nabla_{s-1} + \frac{a_s}{KA_s}\sum_{k=1}^{K}\nabla_{s;k}$, the convexity of g, and $A_0 = 0$, we have

$$\begin{aligned}
\sum_{s=2}^{S}\sum_{k=1}^{K} a_s g(v_{s;k}) \quad & \sum_{s=2}^{S} Ka_s g\Big(\frac{1}{K}\sum_{k=1}^{K} v_{s;k}\Big) \quad \sum_{s=2}^{S}(KA_s g(\nabla_s) \quad KA_{s-1}g(\nabla_{s-1})) \\
&= KA_S g(\nabla_S) \quad KA_1 g(\nabla_1) \\
&= KA_S g(\nabla_S) \quad KA_1 g(v_{1;1}).
\end{aligned} \tag{27}$$

Thus, combining Eqs. (24)–(27), we have

$$\begin{aligned} KA_S f(u) \quad & \frac{K(1+A_S\gamma)}{2}ku \quad v_{S;K}k^2 \quad \frac{K}{2}kx_0 \quad uk^2 + \frac{K}{4}kv_{1;1} \quad v_{1;0}k^2 + Ka_1 f(v_{1;1}) \\ & + \sum_{s=2}^{S}\sum_{k=1}^{K}\Big(a_s h\, r\, f(x_{s;k}) \quad q_{s;k}, u \quad x_{s;k}i + \frac{K(1+A_{s\ 1}\gamma)}{2}kv_{s;k} \quad v_{s;k\ 1}k^2\Big) \\ & + \sum_{s=2}^{S}\sum_{k=1}^{K} a_s(f(x_{s;k}) + hq_{s;k}, v_{s;k} \quad x_{s;k}i) + KA_S g(y_S). \end{aligned} \tag{28}$$

Then with $A_0 = 0$ and $y_s = \frac{1}{K}\sum_{k=1}^{K} y_{s;k}$, we also have

$$\begin{aligned} KA_S f(y_S) = KA_1 f(y_1) + K\sum_{s=2}^{S}(A_s f(y_s) \quad A_{s\ 1} f(y_{s\ 1})) \\ Ka_1 f(v_{1;1}) + \sum_{s=2}^{S}\sum_{k=1}^{K} A_s f(y_{s;k}) \quad K\sum_{s=2}^{S} A_{s\ 1} f(y_{s\ 1}), \end{aligned} \tag{29}$$

where the last equality is by $A_1 = a_1, y_1 = v_{1;1}$. Subtracting Eq. (28) from (29) and noting that $f(y_S) = f(y_S) + g(y_S)$ now completes the proof. □

Lemma 7. *The error sequence $\{E_{s;k}(u)\}_{s\ 2;k2[K]}$ in Lemma 5 satisfies*

$$\begin{aligned} E_{s;k}(u) \quad & A_{s\ 1}(f(y_{s\ 1}) \quad f(x_{s;k}) \quad h\, r\, f(x_{s;k}), y_{s\ 1} \quad x_{s;k}i) \\ & + a_s h\, r\, f(x_{s;k}) \quad q_{s;k}, v_{s;k} \quad ui + \Big(\frac{La_s^2}{2A_s} \quad \frac{K(1+A_{s\ 1}\gamma)}{2}\Big)kv_{s;k} \quad v_{s;k\ 1}k^2. \end{aligned}$$

Proof. Using Assumption 3, Lemma 4, and Lemma 1, and applying the definition of $y_{s;k}$, we have

$$\begin{aligned} & f(y_{s;k}) \quad f(x_{s;k}) \\ & \quad h\, r\, f(x_{s;k}), y_{s;k} \quad x_{s;k}i + \frac{L}{2}ky_{s;k} \quad x_{s;k}k^2 \\ & = h\, r\, f(x_{s;k}), \frac{A_{s\ 1}}{A_s}y_{s\ 1} + \frac{a_s}{A_s}v_{k;s} \quad x_{s;k}i + \frac{La_s^2}{2A_s^2}kv_{s;k} \quad v_{s;k\ 1}k^2 \\ & = \frac{A_{s\ 1}}{A_s}h\, r\, f(x_{s;k}), y_{s\ 1} \quad x_{s;k}i + \frac{a_s}{A_s}h\, r\, f(x_{s;k}), v_{s;k} \quad x_{s;k}i + \frac{La_s^2}{2A_s^2}kv_{s;k} \quad v_{s;k\ 1}k^2. \end{aligned} \tag{30}$$

It remains to plug Eq. (30) into the definition of $E_{s;k}(u)$, and rearrange. □

The definition of the variance reduced extrapolation point $q_{s;k}$ is crucial for bounding the error terms $\{E_{s;k}(u)\}$ from Lemma 5. The next three auxiliary lemmas apply the definition of $q_{s;k}^{(j)}$ to bound the inner product term $h\, r\, f(x_{s;k})$ $q_{s;k}, v_{s;k} \quad ui$ in $E_{s;k}(u)$ when we take the expectation over all randomness in the algorithm. We will use $F_{s;k;i}$ to denote the natural filtration, containing all randomness up to and including epoch s, outer iteration k, and inner iteration i. Note that in Algorithm 4, the index of the inner iteration goes from $j = m$ to 1, therefore inner iteration i corresponds to when index of the inner iteration is $j = m \quad i + 1$. This detail however does not play a important role in our analysis.

Lemma 8. *For all* $s \geq 2$, $k \in [K]$ *and* $u \in \mathrm{dom}(g)$, *we have*

$$
\begin{aligned}
&a_s \mathbb{E}[h \nabla f(x_{s;k}) - q_{s;k}, v_{s;k} - u\rangle] \\
&= \sum_{j=1}^{m} a_s \mathbb{E}[h \nabla^{(j)} f(x_{s;k}) - \nabla^{(j)} f(w_{s;k;j}), v_{s;k}^{(j)} - u^{(j)}\rangle] - \sum_{j=1}^{m} a_{s;k-1} \mathbb{E}[h \nabla^{(j)} f(x_{s;k-1}) - \nabla^{(j)} f(w_{s;k-1;j}), v_{s;k-1}^{(j)} - u^{(j)}\rangle] \\
&\quad + \sum_{j=1}^{m} a_{s;k-1} \mathbb{E}[h \nabla^{(j)} f_{t_j}(x_{s;k-1}) - \nabla^{(j)} f_{t_j}(w_{s;k-1;j}), v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\rangle] \\
&\quad + \sum_{j=1}^{m} a_s \mathbb{E}[h \nabla^{(j)} f(w_{s;k;j}) - (\nabla^{(j)} f_{t_j}(w_{s;k;j}) - \nabla^{(j)} f_{t_j}(\bar{x}_{s-1}) + \bar{\nabla}_s^{(j)}), v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\rangle],
\end{aligned}
$$

Proof. Using the definition of $q_{s;k}^{(j)}$, we have

$$
\begin{aligned}
&a_s\langle \nabla^{(j)} f(x_{s;k}) - q_{s;k}^{(j)}\rangle \\
&= a_s\langle \nabla^{(j)} f(x_{s;k}) - \nabla^{(j)} f(w_{s;k;j})\rangle + a_s\langle \nabla^{(j)} f(w_{s;k;j}) - q_{s;k}^{(j)}\rangle \\
&= a_s\langle \nabla^{(j)} f(x_{s;k}) - \nabla^{(j)} f(w_{s;k;j})\rangle + a_s\langle \nabla^{(j)} f(w_{s;k;j}) - (\nabla^{(j)} f_{t_j}(w_{s;k;j}) - \nabla^{(j)} f_{t_j}(\bar{x}_{s-1}) + \bar{\nabla}_s^{(j)})\rangle \\
&\quad - a_{s;k-1}\langle \nabla^{(j)} f(x_{s;k-1}) - \nabla^{(j)} f_{t_j}(w_{s;k-1;j})\rangle.
\end{aligned} \tag{31}
$$

First, for $j \in [m]$ and any fixed $u^{(j)}$, we have

$$
\begin{aligned}
&\mathbb{E}[a_s h \nabla^{(j)} f(w_{s;k;j}) - (\nabla^{(j)} f_{t_j}(w_{s;k;j}) - \nabla^{(j)} f_{t_j}(\bar{x}_{s-1}) + \bar{\nabla}_s^{(j)}), v_{s;k}^{(j)} - u^{(j)}\rangle] \\
&= \mathbb{E}[a_s h \nabla^{(j)} f(w_{s;k;j}) - (\nabla^{(j)} f_{t_j}(w_{s;k;j}) - \nabla^{(j)} f_{t_j}(\bar{x}_{s-1}) + \bar{\nabla}_s^{(j)}), v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\rangle] \\
&\quad + a_s \mathbb{E}[h\mathbb{E}[\nabla^{(j)} f(w_{s;k;j}) - (\nabla^{(j)} f_{t_j}(w_{s;k;j}) - \nabla^{(j)} f_{t_j}(\bar{x}_{s-1}) + \bar{\nabla}_s^{(j)})|F_{s;k;j-1}], v_{s;k-1}^{(j)} - u^{(j)}\rangle] \\
&= \mathbb{E}[a_s h \nabla^{(j)} f(w_{s;k;j}) - (\nabla^{(j)} f_{t_j}(w_{s;k;j}) - \nabla^{(j)} f_{t_j}(\bar{x}_{s-1}) + \bar{\nabla}_s^{(j)}), v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\rangle],
\end{aligned} \tag{32}
$$

where the first equality follows from $v_{s;k-1}^{(j)} \in F_{s;k;j-1}$ and the second equality follows from $\mathbb{E}[\nabla^{(j)} f_{t_j}(w_{s;k;j})|F_{s;k;j-1}] = \nabla^{(j)} f(w_{s;k;j})$ and $\mathbb{E}[\nabla^{(j)} f_{t_j}(\bar{x}_{s-1})|F_{s;k;j-1}] = \nabla^{(j)} f(\bar{x}_{s-1}) = \bar{\nabla}_s^{(j)}$. Meanwhile, for $j \in [m]$ and any fixed $u^{(j)}$, we have

$$
\begin{aligned}
&\mathbb{E}[a_{s;k-1} h \nabla^{(j)} f_{t_j}(x_{s;k-1}) - \nabla^{(j)} f_{t_j}(w_{s;k-1;j}), v_{s;k}^{(j)} - u^{(j)}\rangle] \\
&= \mathbb{E}[a_{s;k-1} h \nabla^{(j)} f_{t_j}(x_{s;k-1}) - \nabla^{(j)} f_{t_j}(w_{s;k-1;j}), v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\rangle] \\
&\quad + \mathbb{E}[\mathbb{E}[a_{s;k-1} h \nabla^{(j)} f_{t_j}(x_{s;k-1}) - \nabla^{(j)} f_{t_j}(w_{s;k-1;j}), v_{s;k-1}^{(j)} - u^{(j)}\rangle|F_{s;k;j-1}]] \\
&= \mathbb{E}[a_{s;k-1} h \nabla^{(j)} f_{t_j}(x_{s;k-1}) - \nabla^{(j)} f_{t_j}(w_{s;k-1;j}), v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\rangle] \\
&\quad + \mathbb{E}[a_{s;k-1} h \nabla^{(j)} f(x_{s;k-1}) - \nabla^{(j)} f(w_{s;k-1;j}), v_{s;k-1}^{(j)} - u^{(j)}\rangle],
\end{aligned} \tag{33}
$$

where the last equality is by $v_{s;k-1}^{(j)} \in F_{s;k;j-1}$, $\mathbb{E}[\nabla^{(j)} f_{t_j}(x_{s;k-1})|F_{s;k;j-1}] = \nabla^{(j)} f(x_{s;k-1})$ and $\mathbb{E}[\nabla^{(j)} f_{t_j}(w_{s;k-1;j})|F_{s;k;j-1}] = \nabla^{(j)} f(w_{s;k-1;j})$. Combining Eqs. (31)–(33) completes the proof. □

In the following two lemmas, we will bound the third and the fourth terms of the R.H.S. in Lemma 8 by above using our novel Lipschitz Assumption 2 and Assumption 3.

Lemma 9. *For $s \geq 2$ and $k \in [K]$, we have*

$$\sum_{j=1}^{m} a_{s;k-1}\mathbb{E}\left[\left\langle r^{(j)} f_{t_j}(x_{s;k-1}) - r^{(j)} f_{t_j}(w_{s;k-1;j}), v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\right\rangle\right]$$
$$\mathbb{E}\left[\frac{K(1+A_{s-1}\gamma)}{8}\|v_{s;k} - v_{s;k-1}\|^2 + \frac{a_{s;k-1}^4 L^2}{KA_{s;k-1}^2(1+A_{s-1}\gamma)}\|v_{s;k-1} - v_{s;k-2}\|^2\right],$$

where $a_{s;0} = a_{s-1}$, $A_{s;0} = A_{s-1}$ and $a_{s;k} = a_s$, $A_{s;k} = A_s$ for $k \in [K]$.

Proof. Using Cauchy–Schwarz and Young's inequalities, we have

$$\begin{aligned}
&a_{s;k-1}\mathbb{E}\left[\left\langle r^{(j)} f_{t_j}(x_{s;k-1}) - r^{(j)} f_{t_j}(w_{s;k-1;j}), v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\right\rangle\right] \\
&\mathbb{E}\left[\frac{2a_{s;k-1}^2}{K(1+A_{s-1}\gamma)}\left\|r^{(j)} f_{t_j}(x_{s;k-1}) - r^{(j)} f_{t_j}(w_{s;k-1;j})\right\|^2 + \frac{K(1+A_{s-1}\gamma)}{8}\left\|v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\right\|^2\right] \\
&\mathbb{E}\left[\frac{2a_{s;k-1}^2}{K(1+A_{s-1}\gamma)}\|x_{s;k-1} - w_{s;k-1;j}\|_{Q^j}^2 + \frac{K(1+A_{s-1}\gamma)}{8}\left\|v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\right\|^2\right] \\
&= \mathbb{E}\left[\frac{2a_{s;k-1}^2}{K(1+A_{s-1}\gamma)}\|x_{s;k-1} - y_{s;k-1}\|_{(Q^j)_{j+1}}^2 + \frac{K(1+A_{s-1}\gamma)}{8}\left\|v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\right\|^2\right],
\end{aligned} \tag{34}$$

where we used Assumption 3 in the first inequality and the definitions of $x_{s;k-1}$ and $w_{s;k-1;j}$ in the last equality. Finally by including the summation and using the definition of L, $x_{s;k-1}$ and $y_{s;k-1}$, the first term of the above expression becomes

$$\sum_{j=1}^{m}\mathbb{E}\left[\frac{2a_{s;k-1}^2}{K(1+A_{s-1}\gamma)}\|x_{s;k-1} - y_{s;k-1}\|_{(Q^j)_{j+1}}^2\right] = \mathbb{E}\left[\frac{2a_{s;k-1}^2}{K(1+A_{s-1}\gamma)}\|x_{s;k-1} - y_{s;k-1}\|_{\sum_{j=1}^m (Q^j)_{j+1}}^2\right]$$
$$\mathbb{E}\left[\frac{a_{s;k-1}^4 L^2}{KA_{s;k-1}^2(1+A_{s-1}\gamma)}\|v_{s;k-1} - v_{s;k-2}\|^2\right], \tag{35}$$

where $a_{s;0} = a_{s-1}$, $A_{s;0} = A_{s-1}$ and $a_{s;k} = a_s$, $A_{s;k} = A_s$ for $k \in [K]$. Taking summation over j and combining Eqs. (34) and (35) give the lemma statement.

□

Lemma 10. *For $s \geq 2$ and $k \in [K]$, we have*

$$\sum_{j=1}^{m} a_s\mathbb{E}\left[\left\langle r^{(j)} f(w_{s;k;j}) - \left(r^{(j)} f_{t_j}(w_{s;k;j}) - r^{(j)} f_{t_j}(y_{s-1}) + r^{(j)} f(y_{s-1})\right), v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\right\rangle\right]$$
$$\mathbb{E}\left[\left(\frac{2L^2 a_s^4}{KA_s^2(1+A_{s-1}\gamma)} + \frac{K(1+A_{s-1}\gamma)}{8}\right)\|v_{s;k} - v_{s;k-1}\|^2\right]$$
$$+ \frac{8a_s^2 L}{K(1+A_{s-1}\gamma)}\mathbb{E}[f(y_{s-1}) - f(x_{s;k}) - \langle \nabla f(x_{s;k}), y_{s-1} - x_{s;k}\rangle]$$

Proof. Using similar arguments in the proof of Lemma 9, we have

$$a_s\mathbb{E}\left[\left\langle r^{(j)} f(w_{s;k;j}) - \left(r^{(j)} f_{t_j}(w_{s;k;j}) - r^{(j)} f_{t_j}(y_{s-1}) + r^{(j)} f(y_{s-1})\right), v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\right\rangle\right]$$
$$\mathbb{E}\left[\frac{2a_s^2}{K(1+A_{s-1}\gamma)}\left\|r^{(j)} f(w_{s;k;j}) - \left(r^{(j)} f_{t_j}(w_{w;k;j}) - r^{(j)} f_{t_j}(y_{s-1}) + \nabla_s^{(j)}\right)\right\|^2\right.$$
$$\left. + \frac{K(1+A_{s-1}\gamma)}{8}\left\|v_{s;k}^{(j)} - v_{s;k-1}^{(j)}\right\|^2\right]$$

$$= \mathsf{E}\left[\frac{2a_s^2}{K(1+A_{s-1}\gamma)}\mathsf{E}\left[\left\|\left(\nabla^{(j)} f_{t_j}(w_{s;k;j}) + \nabla^{(j)} f_{t_j}(y_{s-1})\right) \quad \left(\nabla^{(j)} f(w_{s;k;j}) \quad {}^{(j)}_{s}\right)\right\|^2 \mid F_{s;k;j-1}\right]\right]$$
$$+ \mathsf{E}\left[\frac{K(1+A_{s-1}\gamma)}{8}\left\|v_{s;k}^{(j)} \quad v_{s;k-1}^{(j)}\right\|^2\right]$$
$$= \mathsf{E}\left[\frac{2a_s^2}{K(1+A_{s-1}\gamma)}\mathsf{E}\left[\left\|\nabla^{(j)} f_{t_j}(w_{s;k;j}) \quad \nabla^{(j)} f_{t_j}(y_{s-1})\right\|^2 \mid F_{s;k;j-1}\right]\right] + \mathsf{E}\left[\frac{K(1+A_{s-1}\gamma)}{8}\left\|v_{s;k}^{(j)} \quad v_{s;k-1}^{(j)}\right\|^2\right]$$
$$\mathsf{E}\left[\frac{4a_s^2}{K(1+A_{s-1}\gamma)}\left(\left\|\nabla^{(j)} f_{t_j}(w_{s;k;j}) \quad \nabla^{(j)} f_{t_j}(x_{s;k})\right\|^2 + \left\|\nabla^{(j)} f_{t_j}(x_{s;k}) \quad \nabla^{(j)} f_{t_j}(y_{s-1})\right\|^2\right)\right. \tag{36}$$
$$\left. + \frac{K(1+A_{s-1}\gamma)}{8}\left\|v_{s;k}^{(j)} \quad v_{s;k-1}^{(j)}\right\|^2\right],$$

where the first equality comes from $\mathsf{E}\left[\nabla^{(j)} f_{t_j}(w_{s;k;j}) \quad \nabla^{(j)} f_{t_j}(y_{s-1}) \mid F_{s;k;j-1}\right] = \nabla^{(j)} f(w_{s;k;j}) \quad \nabla^{(j)} f(y_{s-1})$ since the only randomness is in t_j when conditioned at $F_{s;k;j-1}$, and the last inequality comes from $(a+b)^2 \quad 2(a^2+b^2)$. In order to bound the second term in Eq. (36), we will include the outer summation with respect to j and apply the results from Lemma 1 to get

$$\sum_{j=1}^{m}\mathsf{E}\left[\left\|\nabla^{(j)} f_{t_j}(x_{s;k}) \quad \nabla^{(j)} f_{t_j}(y_{s-1})\right\|^2\right] = \sum_{j=1}^{m}\mathsf{E}\left[\mathsf{E}\left[\left\|\nabla^{(j)} f_{t_j}(x_{s;k}) \quad \nabla^{(j)} f_{t_j}(y_{s-1})\right\|^2 \mid F_{s;k;0}\right]\right]$$
$$= \mathsf{E}\left[\sum_{j=1}^{m}\sum_{l=1}^{n}\frac{1}{n}\left\|\nabla^{(j)} f_l(x_{s;k}) \quad \nabla^{(j)} f_l(y_{s-1})\right\|^2\right]$$
$$= \mathsf{E}\left[\sum_{l=1}^{n}\frac{1}{n}\left\|\nabla f_l(x_{s;k}) \quad \nabla f_l(y_{s-1})\right\|^2\right]$$
$$\mathsf{E}\left[2L\left(f(y_{s-1}) \quad f(x_{s;k}) \quad \langle \nabla f(x_{s;k}), y_{s-1} \quad x_{s;k}\rangle\right)\right], \tag{37}$$

where the second equality comes from $x_{s;k}, y_{s-1} \in F_{s;k;0}$ and the last inequality is by applying Lemma 1 and the definition of $f(x) = \frac{1}{n}\sum_{l=1}^{n} f_l(x)$. To bound the first term of Eq. (36), we apply similar arguments as in Lemma 9 and get

$$\sum_{j=1}^{m}\mathsf{E}\left[\left\|\nabla^{(j)} f_{t_j}(w_{s;k;j}) \quad \nabla^{(j)} f_{t_j}(x_{s;k})\right\|^2\right] \quad \mathsf{E}\left[\frac{a_s^2 L^2}{2A_s^2}\left\|v_{s;k} \quad v_{s;k-1}\right\|^2\right]. \tag{38}$$

Combining Eqs. (36) – (38) gives the lemma statement.

□

Lemma 6. *With* $a_s^2 \quad \frac{KA_{s-1}(1+A_{s-1})}{8L}$, $a_{s;k} = a_s$ *and* $A_{s;k} = A_s$ *for* $k \in [K]$, $a_{s;0} = a_{s-1}$ *and* $A_{s;0} = A_{s-1}$, *then for any fixed* $u \in \mathrm{dom}(g)$ *we have*

$$\sum_{s=2}^{S}\sum_{k=1}^{K}\mathsf{E}[E_{s;k}(u)]$$
$$\sum_{j=1}^{m} a_1\left\langle \nabla^{(j)} f(x_{1;1}) \quad \nabla^{(j)} f(w_{1;1;j}), v_{1;1}^{(j)} \quad u^{(j)}\right\rangle + \sum_{j=1}^{m} a_S\mathsf{E}\left[\left\langle \nabla^{(j)} f(x_{S;K}) \quad \nabla^{(j)} f(w_{S;K;j}), v_{S;K}^{(j)} \quad u^{(j)}\right\rangle\right]$$
$$+ \frac{K}{64}\left\|v_{1;1} \quad v_{1;0}\right\|^2 \quad \frac{5K(1+A_{S-1}\gamma)}{32}\mathsf{E}\left[\left\|v_{S;K} \quad v_{S;K-1}\right\|^2\right],$$

where $x_{1;1}, v_{1;0} \in \mathrm{dom}(g)$ *can be chosen arbitrarily and* $w_{1;1;j}$ *is defined in Algorithm 4.*

Proof. Combining Lemma 7, 8, 9, 10, setting a_s such that $a_s^2 = \frac{KA_{s-1}(1+A_{s-1})}{8L}$ and using $\frac{A_{s-1}}{A_s} \leq 1$, we have

$$\begin{aligned}\mathbb{E}[E_{s;k}(u)] \quad & \sum_{j=1}^{m} a_s \mathbb{E}\left[\left\langle r^{(j)} f(x_{s;k}) - r^{(j)} f(w_{s;k;j}), v_{s;k}^{(j)} - u^{(j)} \right\rangle\right] \\ & \sum_{j=1}^{m} a_{s;k-1} \mathbb{E}\left[\left\langle r^{(j)} f(x_{s;k-1}) - r^{(j)} f(w_{s;k-1;j}), v_{s;k-1}^{(j)} - u^{(j)} \right\rangle\right] \\ & \left(\frac{5K(1+A_{s-1}\gamma)}{32}\right) \mathbb{E}\left[\|v_{s;k} - v_{s;k-1}\|^2\right] + \left(\frac{a_{s;k-1}^4 L^2}{KA_{s;k-1}^2(1+A_{s-1}\gamma)}\right) \mathbb{E}\left[\|v_{s;k-1} - v_{s;k-2}\|^2\right].\end{aligned}$$

Next, by setting $a_{s;0} = a_{s-1}$ and $a_{s;k} = a_s$ for $k = [K]$, we can telescope the error terms and get

$$\begin{aligned}\sum_{s=2}^{S}\sum_{k=1}^{K} \mathbb{E}[E_{s;k}(u)] \quad & \sum_{j=1}^{m}\sum_{s=2}^{S}\sum_{k=1}^{K} a_s \mathbb{E}\left[\left\langle r^{(j)} f(x_{s;k}) - r^{(j)} f(w_{s;k;j}), v_{s;k}^{(j)} - u^{(j)} \right\rangle\right] \\ & \sum_{j=1}^{m}\sum_{s=2}^{S}\sum_{k=1}^{K} a_{s;k-1} \mathbb{E}\left[\left\langle r^{(j)} f(x_{s;k-1}) - r^{(j)} f(w_{s;k-1;j}), v_{s;k-1}^{(j)} - u^{(j)} \right\rangle\right] \\ & \sum_{s=2}^{S}\sum_{k=1}^{K} \frac{5K(1+A_{s-1}\gamma)}{32} \mathbb{E}\left[\|v_{s;k} - v_{s;k-1}\|^2\right] \\ & + \sum_{s=2}^{S}\left[\frac{K(1+A_{s-2}\gamma)}{64}\|v_{s;0} - v_{s;-1}\|^2 + \sum_{k=2}^{K}\frac{K(1+A_{s-1}\gamma)}{64}\|v_{s;k-1} - v_{s;k-2}\|^2\right] \\ & \sum_{j=1}^{m} a_S \mathbb{E}\left[\left\langle r^{(j)} f(x_{S;K}) - r^{(j)} f(w_{S;K;j}), v_{S;K}^{(j)} - u^{(j)} \right\rangle\right] \\ & \sum_{j=1}^{m} a_1 \mathbb{E}\left[\left\langle r^{(j)} f(x_{1;1}) - r^{(j)} f(w_{1;1;j}), v_{1;1}^{(j)} - u^{(j)} \right\rangle\right] \\ & + \frac{K(1+A_0\gamma)}{64}\mathbb{E}\left[\|v_{2;0} - v_{2;-1}\|^2\right] \quad \frac{5K(1+A_{S-1}\gamma)}{32}\mathbb{E}\left[\|v_{S;K} - v_{S;K-1}\|^2\right].\end{aligned}$$

The lemma follows by setting $A_0 = 0$, $v_{2;-1} = v_{1;0}$, $x_{2;0} = x_{1;1}$ and $w_{2;0;j} = w_{1;1;j}$. □

Theorem 2. *Let $x_0 \in \mathrm{dom}(g)$ be an arbitrary initial point. Fix $K \geq 1$ and consider the updates in Algorithm 4. Then for $S \geq 2$ and $\forall u \in \mathrm{dom}(g)$, we have*

$$\mathbb{E}\left[f(y_S) - f(u)\right] + \frac{9(1+A_{S-1}\gamma)}{64A_S}\mathbb{E}\left[\|v_{S;K} - u\|^2\right] \leq \frac{5}{8A_S}\|x_0 - u\|^2.$$

In particular if $x^* = \arg\min_x f(x)$ exists, then we have

$$\mathbb{E}\left[f(y_S) - f(x^*)\right] \leq \frac{5}{8A_S}\|x_0 - x^*\|^2$$

and

$$\mathbb{E}\left[\|v_{S;K} - x^*\|^2\right] \leq \frac{40}{9(1+A_{S-1}\gamma)}\|x_0 - x^*\|^2.$$

Finally in all the bounds above we have

$$A_S \geq \max\left\{\frac{S^2K}{64L}, \frac{1}{4L}\left(1+\sqrt{\frac{K\gamma}{8L}}\right)^{S-1}\right\}.$$

Proof. Combining Lemma 5 and Lemma 6, and by setting $y_{1;0} = x_{1;1} = x_0$ and $y_{1;1} = v_{1;1}$, we have

$$\begin{aligned} KA_S\mathbb{E}\left[f(\hat{y}_S) - f(u)\right] \le{}& \frac{K}{2}\|x_0 - u\|^2 - \frac{K(1+A_S)}{2}\mathbb{E}\left[\|v_{S;K} - u\|^2\right] \\ &- \frac{15K}{64}\|v_{1;1} - x_0\|^2 - \frac{5K(1+A_{S-1}\gamma)}{32}\mathbb{E}\left[\|v_{S;K} - v_{S;K-1}\|^2\right] \\ &+ \sum_{j=1}^{m} a_1\left\langle \nabla^{(j)} f(x_{1;1}) - \nabla^{(j)} f(w_{1;1;j}), v_{1;1}^{(j)} - u^{(j)}\right\rangle \\ &+ \sum_{j=1}^{m} a_S\mathbb{E}\left[\left\langle \nabla^{(j)} f(x_{S;K}) - \nabla^{(j)} f(w_{S;K;j}), v_{S;K}^{(j)} - v_{S;K-1}^{(j)}\right\rangle\right]. \end{aligned} \tag{39}$$

Using the same approach as Lemma 9 and Lemma 10, we can upper bound the first inner product term by

$$\begin{aligned} \sum_{j=1}^{m} a_1\left\langle \nabla^{(j)} f(x_{1;1}) - \nabla^{(j)} f(w_{1;1;j}), v_{1;1}^{(j)} - u^{(j)}\right\rangle &\le \frac{1}{8K}\|v_{1;1} - x_0\|^2 + \frac{K}{16}\|v_{1;1} - u\|^2 \\ &\le \frac{15K}{64}\|v_{1;1} - x_0\|^2 + \frac{K}{8}\|x_0 - u\|^2, \end{aligned} \tag{40}$$

where we used $(a+b)^2 \le 2(a^2+b^2)$, $a_1 = \frac{1}{4L}$ and $K \ge 2$ in the last inequality. Similarly, we have

$$\begin{aligned} &\sum_{j=1}^{m} a_S\mathbb{E}\left[\left\langle \nabla^{(j)} f(x_{S;K}) - \nabla^{(j)} f(w_{S;K;j}), v_{S;K}^{(j)} - v_{S;K-1}^{(j)}\right\rangle\right] \\ &\le \frac{K(1+A_{S-1}\gamma)}{8}\mathbb{E}\left[\|v_{S;K} - v_{S;K-1}\|^2\right] + \frac{K(1+A_{S-1}\gamma)}{64}\mathbb{E}\left[\|v_{S;K} - u\|^2\right], \end{aligned} \tag{41}$$

where we also used $a_S^2 = \frac{KA_{S-1}(1+A_{S-1})}{8L}$ here. Combining Eqs. (39)–(41) gives us our main bounds in the theorem. Lastly, recall that $\{a_s g_{s-1}\}$ is chosen so that $a_s^2 = \frac{KA_{s-1}(1+A_{s-1})}{8L}$. When $\gamma = 0$, this leads to the standard $A_s \ge \frac{k^2K}{64L}$ growth of accelerated algorithms by choosing $a_s = \frac{sK}{32L}$ for $k \ge 1$. When $\gamma > 0$, we have $\frac{a_s}{A_{s-1}} > \sqrt{\frac{K}{8L}}$, and it remains to use that $A_k = \frac{A_k}{A_{k-1}} \cdots \frac{A_2}{A_1} A_1 = A_1\left(1+\sqrt{\frac{K}{8L}}\right)^{k-1}$ where $a_1 = A_1 = \frac{1}{4L}$ using the choice of a_k in Algorithm 1 and $A_0 = a_0 = 0$, completing the proof. □

B.1. Adaptive Variance Reduced A-CODER

Similar to A-CODER, VR-A-CODER can adaptively estimate the Lipschitz parameter by checking the quadratic bounds between $y_{s;}$ and $x_{s;k}$ as well as between $\hat{y}_s$ and $x_{s;k}$. For completeness, we have included the adaptive version of VR-A-CODER in Algorithm 5 below.

Algorithm 5 Variance Reduced A-CODER (Adaptive Version)

Input: $x_0 \in \mathrm{dom}(g), \gamma \geq 0, L_0 > 0, m, \{S^1, \ldots, S^m\}$
Initialization: $\tilde{y}_0 = v_{1;0} = y_{1;0} = x_{1;1} = x_0$; $z_{1;0} = 0$
$L_1 = L_0/2$
repeat
  $L_1 = 2L_1$
  $a_0 = A_0 = 0$; $A_1 = a_1 = \frac{1}{4L_0}$
  $z_{1;1} = \nabla f(x_0)$; $v_{1;1} = \mathrm{prox}_{a_1 g}(x_0 - z_{1;1})$
until $f(v_{1;1}) \leq f(x_0) + \langle \nabla f(x_0), v_{1;1} - x_0 \rangle + \frac{L_1}{2}\|v_{1;1} - x_0\|^2$
$\tilde{y}_1 = y_{1;1} = v_{1;1}$
$w_{1;1;j} = (x_{1;1}^{(1)}, \ldots, x_{1;1}^{(j)}, y_{1;1}^{(j+1)}, \ldots, y_{1;1}^{(m)})$
$v_{2;0} = v_{1;1}$; $w_{2;0;j} = w_{1;1;j}$; $x_{2;0} = x_{1;1}$; $y_{2;0} = y_{1;1}$; $z_{2;0} = z_{1;1}$
for $s = 2$ to S **do**
  $L_s = L_{s-1}/2$
  repeat
    $L_s = 2L_s$
    Set $a_s > 0$ s.t. $a_s^2 = \frac{KA_{s-1}(1+A_{s-1})}{8L_s}$; $A_s = A_{s-1} + a_s$
    $a_{s;0} = a_{s-1}$; $a_{s;1} = a_{s;2} = \cdots = a_{s;K} = a_s$
    $v_{s;0} = v_{s-1;K}$; $w_{s;0;j} = w_{s-1;K;j}$; $x_{s;0} = x_{s-1;K}$; $y_{s;0} = y_{s-1;K}$; $z_{s;0} = z_{s-1;K}$
    $\nabla_s = \nabla f(\tilde{y}_{s-1})$
    for $k = 1$ to K **do**
      $x_{s;k} = \frac{A_{s-1}}{A_s}\tilde{y}_{s-1} + \frac{a_s}{A_s} v_{s;k-1}$
      for $j = m$ to 1 **do**
        $w_{s;k;j} = (x_{s;k}^{(1)}, \ldots, x_{s;k}^{(j)}, y_{s;k}^{(j+1)}, \ldots, y_{s;k}^{(m)})$
        Choose t in $[n]$ uniformly at random
        $\tilde{\tau}_{s;k}^{(j)} = \tau^{(j)} \nabla_t f(w_{s;k;j}) - \tau^{(j)} \nabla_t f(\tilde{y}_{s-1}) + \nabla_s^{(j)}$
        $q_{s;k}^{(j)} = \tilde{\tau}_{s;k}^{(j)} + \frac{a_{s;k-1}}{a_s}(\tau^{(j)} \nabla_t f(x_{s;k-1}) - \tau^{(j)} \nabla_t f(w_{s;k-1;j}))$
        $z_{s;k}^{(j)} = z_{s;k-1}^{(j)} + a_s q_{s;k}^{(j)}$
        $v_{s;k}^{(j)} = \mathrm{prox}_{(A_{s-1} + \frac{a_s k}{K}) g^j}(x_0^{(j)} - z_{s;k}^{(j)}/K)$
        $y_{s;k}^{(j)} = \frac{A_{s-1}}{A_s}\tilde{y}_{s-1}^{(j)} + \frac{a_s}{A_s} v_{s;k}^{(j)}$
      end for
    end for
    $\tilde{y}_s = \frac{1}{K}\sum_{k=1}^{K} y_{s;k}$
  until $f(y_{s;k}) \leq f(x_{s;k}) + \langle \nabla f(x_{s;k}), y_{s;k} - x_{s;k} \rangle + \frac{L_s}{2}\|y_{s;k} - x_{s;k}\|^2$
  and $\frac{1}{n}\sum_{t=1}^{n} \|\nabla f_t(x_{s;k}) - \nabla f_t(\tilde{y}_{s-1})\|^2 \leq 2L_s(f(\tilde{y}_{s-1}) - f(x_{s;k}) - \langle \nabla f(x_{s;k}), \tilde{y}_{s-1} - x_{s;k} \rangle)$
end for
return $v_{S;K}, \tilde{y}_S$