FASTER COORDINATE ALGORITHMS FOR STRUCTURED MACHINE LEARNING PROBLEMS

by

Cheuk Yin (Eric) Lin

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2025

Date of final oral examination: May 07, 2025

The dissertation is approved by the following members of the Final Oral Committee:

Jelena Diakonikolas (Advisor), Assistant Professor, Computer Sciences

Stephen Wright, Professor, Computer Science

James Luedtke , Professor, Industrial and Systems Engineering

Yudong Chen, Associate Professor, Computer Science

Manolis Vlatakis, Assistant Professor, Computer Sciences

To my mother.

ACKNOWLEDGMENTS

I am fortunate to have many people to thank for making my Ph.D. journey and its completion possible. With a bittersweet heart, I make this attempt to thank everyone who has helped and accompanied me along the way.

First and foremost, I would like to thank my advisor, Jelena Diakonikolas. I first met Jelena at her practice job talk on an early Boston morning, despite not being a morning person by any measure, while she was a postdoc and I was a master's student at Boston University. Her talk was about a unified theory of first-order methods in continuous optimization, and it interested me so much that I mustered the rare courage to approach her with my questions after her talk. Little did I know that those few questions initiated what would become many years of collaboration as Jelena's first and probably most difficult-to-advise Ph.D. student ever. Your advice, kindness, and patience have been invaluable throughout this journey. You have been a role model in research and in life, and I still have much to learn from you. I am forever grateful.

I would also like to thank the members of my committee: Stephen Wright, Jim Luedtke, Yudong Chen, and Manolis Vlatakis, for their help in reading this thesis and their valuable feedback, which helped improve this thesis. I would like to especially thank Steve for his generous guidance and the amazing collaborations we had.

I am thankful for having collaborated with many fantastic researchers, whose work I deeply admire and from whom I have learnt about research. I would like to thank Sebastian Pokutta and Alejandro (Alex) Carderera, previously at the Georgia Institute of Technology. I would also like to thank Chaobing Song, who was a postdoc at Madison, with whom we had many fruitful collaborations. I also want to thank Xufeng Cai, who was also a good friend and my office mate for the latter half of

my Ph.D. I also thank Ahmet Alacaoglu for many insightful discussions, even though we have not formally collaborated together.

This journey would not have been possible without the companionship of many peers and friends while at Madison, who all helped keep me motivated in life. I want to express my special thanks to the StudentSay group: Elvis Chang, Kaiyang Chen, Maggie Chen, Yang Guo, Justin LiXie, Holdson Liang, Eric Lin, Jifan Zhang, Xingjian Zhen, and David Zhou. Our diverse perspectives and interests added many colors to my life which I had never experienced before. I also thank the many friends I made during the past few years in Madison through sharing an office, being in the same hallway, or playing badminton together. I apologize for not being able to include everyone, but you all have helped keep me sane.

I would also like to thank the people who have helped bring me to the start of my Ph.D. journey. I am grateful for Prof. Lorenzo Orecchia and Prof. Alina Ene for introducing me to mathematical optimization through their class at Boston University, and for giving me the opportunity to work with them as a research assistant at Boston University. I am also grateful for Prof. Christopher Tout for his guidance and mentorship during and after my studies at the University of Cambridge. I would also like to express my deepest gratitude to my high school headmaster, Mr. Terence Chang, who offered me unconditional support for the unconventional early academic path I had taken.

Finally, and above all, I am forever grateful to my family, and especially to my mother, who raised me single-handedly after my father passed away when I was young. It would not have been possible for me to begin my Ph.D., let alone finish it, without your love, support, and encouragement.

Financial support. The research presented in this thesis was supported in part by NSF grant CCF-2007757 and CCF-2023239, by the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin–Madison with funding from the Wisconsin Alumni Research Foundation, and by the Office of Naval Research under contract number N00014-22-1-2348.

TABLE OF CONTENTS

Page

Page

ABSTRACT

In this dissertation, we study optimization algorithms with coordinate updates, a family of iterative optimization algorithms that serve as the workhorses for training modern data-driven decision systems (also known as machine learning), in stochastic and in cyclic manners. We introduce highly structured related machine learning problems, present several novel algorithms that improve upon existing convergence guarantees, and extend the theoretical understanding of these problems through rigorous mathematical analysis, and detailed numerical experiments. In Chapters 2 and 3, we focus on the cyclic coordinate descent case, where the bias accumulated within each epoch is nontrivial to tackle. In Chapter 4, we shift our focus to solving generalized linear programming (GLP) via randomized coordinate linear primal-dual techniques, and introduce a novel connection between distributionally robust optimization (DRO) and GLP.

Chapter 1

Introduction

Stochastic gradient descent (SGD) and coordinate descent (CD) are the two mostly influential optimization algorithms in the era of modern machine learning and deep learning problems. While both classes of algorithms have a long history in the area of mathematical optimization, their core idea is to reduce full expensive update rules to simple update rules with cheap per-iteration costs. In the modern era where there is a seemingly unlimited supply of data and unimaginably scalable computing resources for training predictive models, their cheap per-iteration costs and lower memory overhead make them particularly attractive as the go-to optimization algorithms. At the same time, their resurgence has motivated a new wave of research into their theoretical properties and their practical performances. In this dissertation, we focus on improving the understanding of coordinate descent algorithms with cyclic and stochastic strategies for solving structured smooth optimization and min-max problems, which have strong connections to many modern machine learning applications. In this chapter, we focus on defining the notation that will be used throughout the thesis, as well as the basics of the algorithms of interest.

1.1 Problem Description

The basic continuous optimization problem can be phrased as

$$\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) + r(\boldsymbol{x}) \tag{1.1}$$

where $\mathcal{X} \subseteq \mathbb{R}^d$ is a compact convex set, $f : \mathbb{R}^d \to \mathbb{R}$ is a convex objective function and $r : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is an optional regularization function. Depending on the structure of the problem, its optimization problem can be of composite form or not, while its objective function can be smooth/nonsmooth and strongly-convex/non-strongly convex. In the following, we give a brief overview of the common properties in convex optimization.

Definition 1 (Convex set)**.** *A set $\mathcal{X}$ is* convex *if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ and $0 \leq \gamma \leq 1$ it holds that*

$$\gamma \boldsymbol{x} + (1-\gamma)\boldsymbol{y} \in \mathcal{X}. \tag{1.2}$$

Definition 2 (Convex function)**.** *A function f is* convex *over $\mathcal{X}$ if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ and $0 \leq \gamma \leq 1$ it holds that*

$$f(\gamma \boldsymbol{x} + (1-\gamma)\boldsymbol{y}) \leq \gamma f(\boldsymbol{x}) + (1-\gamma) f(\boldsymbol{y}). \tag{1.3}$$

We further consider common *nice* properties that facilitate important mathematical analysis on the convergence rates of algorithms under various assumptions. In particular, we define Lipschitz continuity, L-smoothness and μ-strong convexity as follows:

Definition 3 (Lipschitz continuity)**.** *A function f is M-Lipschitz continuous over $\mathcal{X}$ if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ it holds that*

$$\|f(\boldsymbol{x}) - f(\boldsymbol{y})\|_2 \leq M \|\boldsymbol{x} - \boldsymbol{y}\|_2 . \tag{1.4}$$

Definition 4 (Lipschitz smoothness)**.** *A differentiable function f is L-Lipschitz smooth over $\mathcal{X}$ if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ it holds that*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L \|\boldsymbol{x} - \boldsymbol{y}\|_2 . \tag{1.5}$$

Definition 5 (Strong convexity)**.** *A function f is μ-strongly convex over $\mathcal{X}$ if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ there exists $\mu > 0$ and $0 \leq \gamma \leq 1$ such that*

$$f(\gamma \boldsymbol{x} + (1-\gamma)\boldsymbol{y}) \leq \gamma f(\boldsymbol{x}) + (1-\gamma) f(\boldsymbol{y}) - \gamma(1-\gamma)\frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 . \tag{1.6}$$

1.2 Favorable Structures for Efficient Algorithms in Modern Machine Learning Problems

Without the presence of any favorable structures such as the above defined convexity, smoothness or strong convexity, an optimization problem for predictive models can be *difficult*. In fact, a general nonconvex nonsmooth minimization problem for predictive models can be NP-hard with respect to the number of model parameters in the worst case. However in the past few decades, overwhelming empirical evidence in machine learning and deep learning has shown that optimization problems in most model training are not near the difficulty of NP-hardness (e.g., [KSH12]). Therefore it is natural to believe that those problems possess some forms of favorable structures, leading to some popular assumptions of niceness in algorithm designs. In this section, we present two additional structures that motivate the innovation of more efficient and scalable algorithms in this thesis.

1.2.1 Data Separability

Even in modern *big data* applications, it is natural to consider a dataset as a finite collection of data points where each data point (or each batch of data points) can be accessed cheaply and independently. This motivates the following form of data-separable finite-sum structure in our optimization problem:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \Big\{ f(\boldsymbol{x}) + r(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}) + r(\boldsymbol{x}) \Big\} \tag{1.7}$$

where f_i correspond to the loss function with respect to the i-th data point and n is the total number of data points in the dataset. This structure permits the designs of efficient stochastic algorithms by variance reduction techniques.

1.2.2 Coordinate Separability

Modern learning problems are very high-dimensional. In certain structured problems where they can be decomposed into smaller simpler subproblems along the dimension of parameters [PWX+16], we can consider each update on one or a small block of parameters while fixing others. This observation motivates us to partition $\boldsymbol{x}$ into coordinate blocks $(x^1, x^2, ..., x^d)$ where d is the number of model parameters.

These types of coordinate algorithms are particularly useful when the data matrix is sparse, leading to cheap updates when solving coordinate subproblems during each coordinate update. In recent years, we have also seen a renewed interest to model distributed large language model training as coordinate separable problem to understand the parallelization of such algorithms.

1.3 Contributions and Organization

In the following, I briefly outline the organization of this thesis and our contributions. I defer the details of background-related work for each topic to its respective chapter due to the diversity of areas.

Chapter 1: In the first chapter we revisit fundamental concepts in convex optimization that will be used throughout the thesis. We also define the optimization problem and discuss its characteristic and structures.

Chapter 2: Shuffled-style Stochastic gradient descent (SGD) algorithms are a class of simple yet powerful algorithms that are widely used in practice to train large-scale deep learning models. However, in contrast to its widely-studied theoretical counterpart which usually relies on the assumption of sampling with replacement, the convergence behavior of shuffled-style SGD is much less well-understood despite its superior numerical performance. We propose to view shuffled-style SGD in a primal-dual perspective where the gradient descent iterations with respect to individual samples in datasets becomes cyclic coordinate steps in the dual space. We show improved fine-grained convergence bounds and offer a theoretical explanation of the superior empirical performance of data permutations over vanilla counterparts in machine learning problems.

Chapter 3: Traditional gradient descent algorithms solve optimization problems by iteratively computing the full gradients and perform updates over all model parameters at each step. In contrast, coordinate descent algorithms solve them by successively performing minimization along coordinate directions or hyperplanes (in block coordinate setting). We focus on accelerating cyclic coordinate algorithms via introducing dual averaging with extrapolation and make an important step towards a dimension-independent algorithm in the class of cyclic algorithms.

Chapter 4: Departing from the cyclic settings of Chapter 2 and 3, we study a class of GLP in a large-scale settings via a form of randomized dual coordinate algorithm. We reformulate GLP as an equivalent convex-concave min-max problem where we exploit the linear structure in the problem and propose an efficient scalable first-order algorithm named CLVR. We introduce a novel reformulation connecting DRO problems with ambiguity sets based on both f-divergence and Wasserstein metrics to GLP, and demonstrate the practical effectiveness in solving DRO problems through CLVR.

1.4 Other Work During My Ph.D.

Some of the work done during my Ph.D. was not included in this thesis because it was not closely related to the theme or because it has not been fully completed yet, but I would like to take the opportunity to mention it in this section.

Parameter-free Locally Accelerated Conditional Gradients. The first project during my Ph.D. involves studying parameter-free acceleration under conditional gradient settings. Projection-free conditional gradient (CG) methods are the algorithms of choice for constrained optimization setups in which projections are often computationally prohibitive but linear optimization over the constraint set remains computationally feasible. Unlike in projection-based methods, globally accelerated convergence rates are in general unattainable for CG. However, a very recent work (at

the time) on Locally accelerated CG (LaCG) has demonstrated that local acceleration for CG is possible for many settings of interest. The main downside of LaCG is that it requires knowledge of the smoothness and strong convexity parameters of the objective function. We remove this limitation by introducing a novel, Parameter-Free Locally accelerated CG (PF-LaCG) algorithm, for which we provide rigorous convergence guarantees. Our theoretical results are complemented by numerical experiments, which demonstrate local acceleration and showcase the practical improvements of PF-LaCG over non-accelerated algorithms, both in terms of iteration count and wall-clock time. Figure 1.1 provides a brief description of the design and execution of PF-LaCG.

Efficient Waveform De-convolution for Neutrino Detection. Towards the end of my Ph.D., Jelena and I began a collaboration with wonderful scientists Benedikt Riedel, Jim Braun and Josh Peterson from the IceCube Neutrino Observatory. Despite making significant algorithmic progress, our work has not been deployed to their production yet, thus yet a publication, due to various constraints.

The IceCube Neutrino Observatory is a 1 km^3 neutrino detector located in the South Pole, optimized for detection of high-energy astrophysical neutrinos [A$^+$17]. When a high energy neutrino interacts with a nucleon, ultra-relativistic charged particles are produced and emit Cherenkov radiation. Digital optical modules (DOMs), housing a photomultiplier-tube (PMT), measure and digitize the voltage waveforms produced by incoming Cherenkov photons. The algorithmic challenge arises during the phase where the digitized waveform is required to be de-convoluted into a pulse series with photon charge (approximate number of photons) and timing information. The de-convolution problems can be formulated as a Non-Negative Least Square (NNLS+) problem

$$\min_{\boldsymbol{x}\geq 0} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + r(\boldsymbol{x})$$

where the basis matrix $\boldsymbol{A}$ is non-negative, highly sparse, and highly structured. Furthermore, the desired form of regularization $r(\boldsymbol{x})$ is unclear due to various physical properties which we would like the algorithm to attain. We propose a fast multi-stage block-coordinate exact minimization algorithm which leverages the block incremental structure of the basis matrix to repeatedly solve a simpler approximated subproblem. Furthermore, we show improved computational efficiency in terms of CPU-time compared to the production algorithm, which is a Lawson Hanson NNLS algorithm specialized for use in IceCube. We hope to further optimize our algorithm and aim to have our work deployed in the next generation of IceCube Gen II.

1.5 Contributions to Literature

- The non-included work on Parameter-free Locally Accelerated Conditional Gradients was published in the proceedings of ICML'21 [CDLP21]. Carderera, A., Diakonikolas, J., Lin, C.Y., & Pokutta, S. (2021). Parameter-free Locally Accelerated Conditional Gradients. International Conference on Machine Learning.

- Chapter 2 is based on [CLD24]: Cai, X.*, Lin, C. Y.*, & Diakonikolas, J. (2024). Tighter Convergence Bounds for Shuffled SGD via Primal-Dual Perspective. Advances in Neural Information Processing Systems.

- Chapter 3 is based on [LSD23]: Lin, C.Y., Song, C., & Diakonikolas, J. (2023). Accelerated Cyclic Coordinate Dual Averaging with Extrapolation for Composite Convex Optimization. International Conference on Machine Learning.

- Chapter 4 is based on [SLWD22]: Song, C.*, Lin, C. Y.*, Wright, S., & Diakonikolas, J. (2022). Coordinate Linear Variance Reduction for Generalized Linear Programming. Advances in Neural Information Processing Systems.

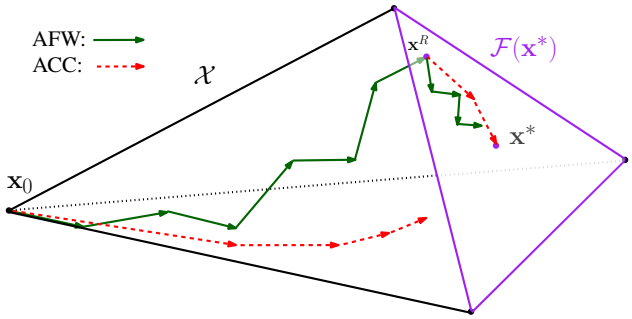


Figure 1.1: An example of coupling between Away-Step Frank Wolfe (AFW) and accelerated gradient descent (ACC) in PF-LaCG on a tetrahedron as the feasible set, starting from initial point $\boldsymbol{x}_0$ with the base of the tetrahedron as its support $\mathcal{S}_0$. The two algorithms are run in parallel from $\boldsymbol{x}_0$: AFW optimizes over the entire tetrahedron, allowing it to add and remove vertices, while ACC optimizes over the base of the tetrahedron only and it cannot converge to the optimal point $\boldsymbol{x}^*$, as $\boldsymbol{x}^* \notin \text{co}(\mathcal{S}_0)$. After several iterations, once the restart criterion for AFW is triggered, PF-LaCG chooses the output point of AFW over that of ACC, as $w^{\text{AFW}} \leq \min\{w^{\text{ACC}}, w^{\text{ACC}}_{\text{prev}}/2\}$, hence a PF-LaCG restart occurs at $\boldsymbol{x}^R$. For ease of exposition we assume that the point outputted by AFW is contained in $\mathcal{F}(\boldsymbol{x}^*)$ after a single halving of $w(\boldsymbol{x}, \mathcal{S})$, although in practice several restarts may be needed for AFW to reach $\mathcal{F}(\boldsymbol{x}^*)$. Since $\boldsymbol{x}^R$ is on the optimal face $\mathcal{F}(\boldsymbol{x}^*)$, PF-LaCG has completed the burn-in phrase. The two algorithms again run in parallel from $\boldsymbol{x}^R$ after the restart. However, ACC converges to the optimal $\boldsymbol{x}^*$ at an accelerated rate, much faster than AFW. Hence, local acceleration is achieved by PF-LaCG while being at least as fast as vanilla AFW.

Chapter 2

Tighter Convergence Bounds for Shuffled SGD via Primal-Dual Perspective

Stochastic gradient descent (SGD) is the most fundamental optimization algorithm in modern machine learning. Similar to the traditional gradient descent algorithm, SGD is an iterative algorithm that updates the model parameters based on the gradient of the loss function, minimizing the loss function in the process. Different than the traditional gradient descent algorithm, SGD approximates the gradient of the loss function by computing it from a single sample or a mini-batch of sample at each step. This is a simple yet powerful algorithm, removing the memory and computation burden of making a full pass over the entire dataset. This feature has allowed SGD, along with its many variants such as momentum SGD, ADAM and RMSProp, to scale particularly well in deep learning applications, where models often have millions or even billions of parameters trained over billions rows of data. Despite its huge success in modern machine learning applications, there exist fundamental gaps between the theoretical understanding of these algorithms vs their practical effectiveness. In this chapter, we will focus on studying a variant of SGD most widely used in practice – Shuffled SGD.

The most basic version of SGD picks a single sample from the dataset at random, computes its gradient and conducts an update to model parameters at each step. The empirical practice instead samples from the dataset *without replacement* (shuffling) and with (possible) reshuffling at each epoch. While this shuffling strategy does not align with the theoretical counterpart of SGD which usually relies on the assumption of *sampling with replacement*, it has proven to be extremely effective in training complex machine learning and deep learning models over large datasets. It is only very recently that SGD using sampling without replacement – shuffled SGD – has been analyzed with

matching upper and lower bounds. However, we observe that those bounds are too pessimistic to explain often the superior empirical performance of data permutations (sampling without replacement) over vanilla counterparts (sampling with replacement) on machine learning problems.

2.1 Introduction

Originally proposed in [RM51], SGD has been broadly studied in the machine learning literature due to its effectiveness in large-scale settings, where full gradient computations are often computationally prohibitive. When applied to unconstrained finite-sum problems

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}), \quad \text{where} \quad f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}), \tag{P}$$

SGD performs the update $\boldsymbol{x}_t = \boldsymbol{x}_{t-1} - \eta\nabla f_{i_t}(\boldsymbol{x}_{t-1})$ for $i_t \in [n]$ ($[n] := \{1,\dots,n\}$), in each iteration t. Traditional theoretical analysis for SGD builds upon the assumption of sampling $i_t \in [n]$ with replacement according to a fixed distribution $\mathbf{p} = (p_1,\dots,p_n)^\top$ over $[n]$, which leads to $\mathbb{E}_{i_t}[\nabla f_{i_t}(\boldsymbol{x}_{t-1})/(np_{i_t})] = \nabla f(\boldsymbol{x}_{t-1})$, and thus much of the (deterministic) gradient descent-style analysis can be transferred to this setting. By contrast, no such connection between the component and the full gradient can be established for shuffled SGD — which employs sampling *without replacement* — making its analysis much more challenging. As a result, despite its fundamental nature, there were no non-asymptotic convergence results for shuffled SGD until a very recent line of work [GOP21, Sha16, HS19, NJN19, RGP20, AYS20, MKR20, NTDP$^+$21, CLY23]. All existing results consider general finite sum problems, with the same regularity condition constant (Lipschitz constant of f_i or its gradient) assumed for all the component functions. As a result, the obtained convergence bounds are typically no better than for (full) gradient descent, and are only better than the bounds for SGD with replacement sampling if the algorithm is run for many full passes over the data [MKR20, NTDP$^+$21].

Furthermore, there is a large gap between the empirical performance of shuffled SGD and the predicted convergence rates from prior work [MKR20, GOP21]. One cause for this discrepancy are overly pessimistic bounds on the step size in prior work, which are of order $1/(nL_{\max})$, where $L_{\max}$ is the maximum smoothness constant over components f_i in (P). In practice, the step sizes are tuned to achieve better convergence bounds than predicted by the current theory. We illustrate how restrictions on the step size affect convergence of shuffled SGD (with random permutations in each epoch) in Fig. 2.1, where we plot the resulting optimality gap over full data passes when shuffled SGD is applied to logistic regression problems on standard datasets. To compare the effect of the step size η from prior work and our work, we choose take $\eta = 1/(\sqrt{2}nL_{\max})$ based on [MKR20], and $\eta = 1/(n\sqrt{\hat{L}\tilde{L}})$ from our work, where $\hat{L}, \tilde{L}$ are our novel fine-grained, data-dependent smoothness

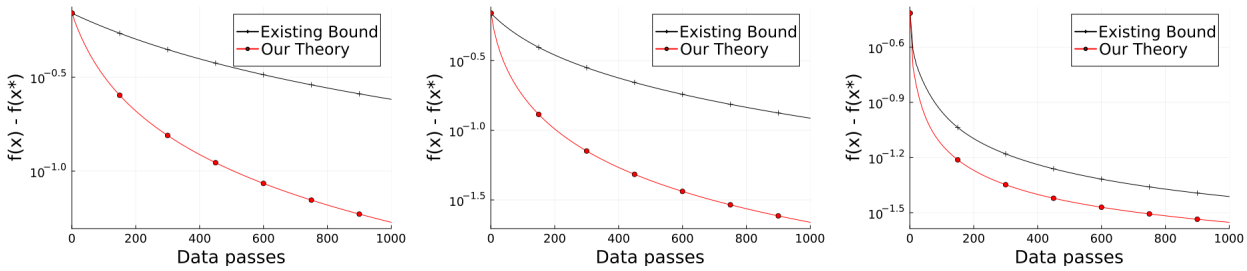


Figure 2.1: An illustration of the convergence behaviour of shuffled SGD for logistic regression problems on LIBSVM datasets `luke`, `leu` and `a9a`, where we use step sizes from existing bounds and our work. Due to randomness, we average over 20 runs for each plot and include a ribbon around each line to show its variance. However, as suggested by the concentration of $\hat{L}$ (see Subsection 2.5.1 and Subsection 2.5.2), the variance across multiple runs is negligible, hence the ribbons are barely observable here. Our finding shows a significant gap between the current theoretical understanding of shuffled SGD (existing bound) and its potential practical performance, motivating us to investigate whether the current bounds are too pessimistic. We observe that our novel bounds can offer up to $\sqrt{\min(n,d)}$ improvement on the step sizes, which translates to the predicted convergence rates compared to state of the art upper bounds.

parameters defined in Section 2.4 for smooth convex finite-sum problems with linear predictors. As can be observed from Fig. 2.1, larger step sizes resulting from our theory lead to faster convergence of shuffled SGD and, as a result, our convergence bounds better predict the performance of shuffled SGD.

Building on these insights, we introduce a fined-grained theoretical analysis to transparently show how the structure of the data and the possibly different Lipschitz constants of the component functions or their gradients affect the performance of shuffled SGD, thus providing a better explanation of the heuristic success of shuffled SGD in modern machine learning.

2.2 Our Results

Through fine-grained analysis in the lens of primal-dual cyclic coordinate methods and the introduction of novel smoothness parameters, we present several results for shuffled SGD on smooth and non-smooth convex losses, where our novel analysis framework provides tighter convergence bounds over all popular shuffling schemes (IG, SO, and RR). Notably, our new bounds predict faster convergence than existing bounds in the literature – by up to a factor of $O(\sqrt{n})$, mirroring benefits from tighter convergence bounds using component smoothness parameters in randomized coordinate

methods. Lastly, we numerically demonstrate on common machine learning datasets that our bounds are indeed much tighter, thus offering a bridge between theory and practice.

In this chapter, we study the convergence rates of shuffled SGD in various settings through a unified primal-dual perspective, making intriguing connections to cyclic coordinate methods. This analysis framework is novel and allows us to leverage cyclic bias accumulation techniques on the dual side to obtain fine-grained convergence bounds. The obtained bounds mirror the improvements in randomized coordinate methods, which come from different coordinate smoothness parameters. While coordinate methods are no better than full-gradient methods in the worst case, on typical problem instances, they are much faster and the improvements come precisely from a more fine-grained view of smoothness. We see a similar phenomenon in our analysis, which highlights the usefulness of the fine-grained smoothness characterizations introduced in our work.

We provide improved bounds for all three popular data permutation strategies RR, SO and IG, in smooth convex settings. When the problem objective narrows to empirical risk minimization with linear predictors, we are able to exploit the data-dependent structure and uncouple the linear and nonlinear parts of the objective function, allowing us to provide tighter data-dependent bounds, up to a factor of $O(\sqrt{n})$. Moreover, we show that our techniques extend to non-smooth convex settings, providing improved bounds over existing work.

We summarize our results and compare them to the state of the art in Table 2.1. As is standard, all complexity results in Table 2.1 are expressed in terms of individual (component) gradient evaluations. They represent the number of gradient evaluations required to construct a solution with (expected) optimality gap ϵ, given a target error $\epsilon > 0$.

Extensions to mini-batching and IG. When presenting our results for general finite-sum problems (in Section 2.3), we consider simple updates without mini-batching for ease of presentation and to avoid introducing excessive notation. However, we emphasize that all our results can be extended to shuffled SGD with mini-batching. Our results are also the first to provide convergence bounds that demonstrate benefits of mini-batching in shuffled SGD. For completeness and generality, the proofs in the appendix are carried out for mini-batch settings with arbitrary batch sizes $b \in \{1, \ldots, n\}$. Thus, all the results stated in Section 2.3 can be recovered by setting $b = 1$. Moreover, our framework can provide similar fine-grained convergence bounds for IG.

2.2.1 Background and related work

SGD (with replacement) has been extensively studied in many settings (see e.g., [RM51, BCN18, B$^+$15, AWBR09] for convex optimization). Compared to SGD, shuffled SGD usually exhibits faster convergence in practice [Bot09, RR13], and is easier and more efficient to implement [Ben12]. For

each epoch k, shuffled SGD-style algorithms perform incremental gradient updates based on the sample ordering (permutation of the data points) denoted by $\pi^{(k)}$. There are three main choices of data permutations: (i) $\pi^{(k)} \equiv \pi$ for some fixed permutation of $[n]$ for all epochs, where shuffled SGD reduces to the incremental gradient (IG) method; (ii) $\pi^{(k)} \equiv \tilde{\pi}$ where $\tilde{\pi}$ is randomly chosen only once, at the beginning of the first epoch, referred to as the shuffle-once (SO) scheme; (iii) $\pi^{(k)}$ randomly generated at the beginning of each epoch, referred to as random reshuffling (RR).

For general smooth convex settings, the convergence of shuffled SGD has been established only recently. For the number of epochs K sufficiently large, [NJN19] proved a convergence rate $\mathcal{O}(1/\sqrt{nK})$ for RR, which leads to the complexity matching SGD. This result was later improved to $\mathcal{O}(1/(n^{1/3}K^{2/3}))$ by [MKR20, NTDP$^+$21, CLY23] for K sufficiently large and with bounded variance assumed at the minimizer, while the same rate holds for SO [MKR20]. These results were complemented by matching lower bounds in [CLY23], under sufficiently small step sizes as utilized in prior work. The results in [MKR20, NTDP$^+$21] require restricted $\mathcal{O}(1/(nL))$ step sizes and reduce to $\mathcal{O}(1/K)$ for small K, acquiring the same iteration complexity as full-gradient methods. Unlike in strongly convex settings, we are not aware of any follow-up work with improvements under small K for smooth convex settings.

The major difficulty in analyzing shuffled SGD comes from characterizing the difference between the intermediate iterate and the iterate after one full data pass, for which current analysis (see e.g., [MKR20] in smooth convex settings) uses the global smoothness constant with a triangle inequality. Such a bound may be too pessimistic and fail capturing the nuances of intermediate progress of shuffled SGD, which leads to a small step size and large K restrictions. To provide a more fine-grained analysis that narrows the theory-practice gap for shuffled SGD, we notice that such a proof difficulty is reminiscent of the analysis of cyclic block coordinate methods relating the partial gradients to the full one. This natural connection was further emphasized in studies of cyclic methods with random permutations [LW19, WL20]; however, these results were limited to convex quadratics. More generally, it is possible to interpret shuffled SGD as a primal-dual method performing cyclic updates on the dual side (see (PD) in Section 2.3.1 and (PL-PD) in Section 2.4). We note here that prior work on dual coordinate methods [SSZ13] provided theoretical guarantees only for the algorithms that choose the dual coordinate to optimize uniformly at random, while the cyclic variant (related to shuffled SGD) had only been studied numerically up until this work.

In this chapter, we view shuffled SGD as a primal-dual method where the updates are performed on the dual side in a cyclic manner, thus we can leverage techniques from general cyclic methods. However, in contrast to randomized methods (corresponding to standard SGD), cyclic methods are usually more challenging to analyze [Nes12], basic variants exhibit much worse *worst-case* complexity than even full gradient methods [LZA$^+$17, SY21, BT13, GOPV17, LZA$^+$17, ST13, XY15, XY17],

Table 2.1: Comparison of our results with state of the art, in terms of individual gradient oracle complexity required to output $\boldsymbol{x}_{\text{out}}$ with $\mathbb{E}[f(\boldsymbol{x}_{\text{out}}) - f(\boldsymbol{x}_*)] \leq \epsilon$, where $\epsilon > 0$ is the target error and $\boldsymbol{x}_*$ is the optimal solution. Here, $\sigma_*^2 = \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_*)\|_2^2$, $D = \|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2$, and generalized linear model refers to objectives of the form $f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n \ell_i(\boldsymbol{a}_i^\top \boldsymbol{x})$ as defined in Section 2.4. Parameters $\hat{L}^g, \tilde{L}^g$ are defined in Section 2.3 and satisfy $\hat{L}^g \leq \frac{1}{n}\sum_{i=1}^n L_i$ and $\tilde{L}^g \leq L_{\max}$. Parameters $\hat{L}$, $\tilde{L}$, and $\bar{G}$ are defined in Section 2.4, and are discussed in the text of this section.

PAPER		COMPLEXITY	ASSUMPTIONS	STEP SIZE
[NTDP+21] [CLY23]	(RR)	$\mathcal{O}\big(\frac{nL_{\max}D^2}{\epsilon} + \frac{\sqrt{nL_{\max}}\sigma_* D^2}{\epsilon^{3/2}}\big)$	f_i: $L_{\max}$-SMOOTH, CONVEX	$\mathcal{O}\big(\frac{1}{nL_{\max}}\big)$
[MKR20]	(RR/SO)	$\mathcal{O}\big(\frac{nL_{\max}D^2}{\epsilon} + \frac{\sqrt{nL_{\max}}\sigma_* D^2}{\epsilon^{3/2}}\big)$	f_i: $L_{\max}$-SMOOTH, CONVEX	$\mathcal{O}\big(\frac{1}{nL_{\max}}\big)$
[Ours, Theorem 1]	(RR/SO)	$\mathcal{O}\big(\frac{n\sqrt{\hat{L}^g\tilde{L}^g}D^2}{\epsilon} + \frac{\sqrt{n\tilde{L}^g}\sigma_* D^2}{\epsilon^{3/2}}\big)$	f_i: L_i-SMOOTH, CONVEX	$\mathcal{O}\big(\frac{1}{n\sqrt{\hat{L}^g\tilde{L}^g}}\big)$
[Ours, Theorem 3]	(RR/SO)	$\mathcal{O}\big(\frac{n\sqrt{\hat{L}\tilde{L}}D^2}{\epsilon} + \frac{\sqrt{n\tilde{L}}\sigma_* D^2}{\epsilon^{3/2}}\big)$	ℓ_i: L_i-SMOOTH, CONVEX GENERALIZED LINEAR MODEL	$\mathcal{O}\big(\frac{1}{n\sqrt{\hat{L}\tilde{L}}}\big)$
[CLY23] LOWER BOUND	(RR)	$\Omega\big(\frac{\sqrt{nL_{\max}}\sigma_* D^2}{\epsilon^{3/2}}\big)$	f_i: $L_{\max}$-SMOOTH, CONVEX, LARGE K	$\mathcal{O}\big(\frac{1}{nL_{\max}}\big)$
[SHA16]	(RR/SO)	$\mathcal{O}\big(\frac{\bar{B}^2 G_{\max}^2}{\epsilon^2}\big)$ $(K=1,\ n=\Omega(1/\epsilon^2))$	ℓ_i: $G_{\max}$-LIPSCHITZ, CONVEX $\bar{B}$-BOUNDED ITERATES, $\|\boldsymbol{a}_i\| \leq 1$ GENERALIZED LINEAR MODEL	$\mathcal{O}\big(\frac{1}{\sqrt{n}}\big)$
[Ours, Theorem 5]	(RR/SO)	$\mathcal{O}\big(\frac{n\bar{G}D^2}{\epsilon^2}\big)$	ℓ_i: G_i-LIPSCHITZ, CONVEX GENERALIZED LINEAR MODEL	$\mathcal{O}\big(\frac{1}{n\sqrt{\bar{G}K}}\big)$

with more refined results being established only recently [SD21b, CSWD22b, LSD23]. While the inspiration for our work came from these recent results [SD21b, CSWD22b, LSD23], they are completely technically disjoint. First, all these results rely on non-standard block Lipschitz assumptions, which are not present in our work. Second, all of them leverage proximal gradient-style cyclic updates to carry out the analysis, which is inapplicable in our case for the cyclic updates on the dual side, as otherwise the method would not correspond to (shuffled) SGD. Finally, [SD21b, LSD23] utilize extrapolation steps, which would break the connection to shuffled SGD in our setting, while [CSWD22b] relies on a gradient descent-type descent lemma, which is impossible to establish in our setting.

2.2.2 Notation and preliminaries

We consider a real d-dimensional Euclidean space $(\mathbb{R}^d, \|\cdot\|)$ where d is finite and $\|\cdot\|$ is the ℓ_2-norm. For a vector $\boldsymbol{x}$, we let $\boldsymbol{x}^j$ denote its j-th coordinate. For any positive integer m, we use $[m]$ to denote the set $\{1, 2, \ldots, m\}$. Given a matrix $\boldsymbol{A}$, $\|\boldsymbol{A}\| := \sup_{\boldsymbol{x}\in\mathbb{R}^d, \|\boldsymbol{x}\|\leq 1} \|\boldsymbol{A}\boldsymbol{x}\|$ denotes its operator norm. For a positive definite matrix $\boldsymbol{\Lambda}$, $\|\cdot\|_{\boldsymbol{\Lambda}}$ denotes the Mahalanobis norm, $\|\boldsymbol{x}\|_{\boldsymbol{\Lambda}} := \sqrt{\langle \boldsymbol{\Lambda}\boldsymbol{x}, \boldsymbol{x}\rangle}$. We use $\boldsymbol{I}$ to denote the identity matrix, and $\operatorname{diag}(\boldsymbol{v})$ to denote the diagonal matrix with vector $\boldsymbol{v}$ on the main diagonal. For any $j \in [n]$, we define $\boldsymbol{I}_{j\uparrow}$ as the matrix obtained from the identity matrix $\boldsymbol{I}$ by setting the first j diagonal elements to zero, and let $\boldsymbol{I}_j$ be the matrix with only the j-th diagonal element nonzero and equal to 1. To handle the cases with random data permutations, we use the following definitions corresponding to the data permutation $\pi = \{\pi^1, \pi^2, \ldots, \pi^n\}$ of $[n]$: $\boldsymbol{A}_\pi := [\boldsymbol{a}_{\pi_1}, \boldsymbol{a}_{\pi_2}, \ldots, \boldsymbol{a}_{\pi_n}]^\top$ permuting the rows based on π given a matrix $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n]^\top$, and $\boldsymbol{v}_\pi := (\boldsymbol{v}^{\pi_1}, \boldsymbol{v}^{\pi_2}, \ldots, \boldsymbol{v}^{\pi_n})^\top$ permuting the coordinates/subvectors based on π given a vector $\boldsymbol{v} = (\boldsymbol{v}^1, \boldsymbol{v}^2, \ldots, \boldsymbol{v}^n)^\top$.

2.3 Primal-Dual Framework for Smooth Convex Finite-Sum Problems

Throughout this section, we make the following standard assumptions.

Assumption 1. *Each f_i is convex and L_i-smooth, and there exists a minimizer $\boldsymbol{x}_* \in \mathbb{R}^d$ for $f(\boldsymbol{x})$.*

Assumption 1 implies that f and all component functions f_i are L-smooth, where $L_{\max} := \max_{i\in[n]} L_i$. It also implies that each convex conjugate f_i^* is $\frac{1}{L_i}$-strongly convex [Bec17]. In this section, we define $\boldsymbol{\Lambda} = \operatorname{diag}(\underbrace{L_1, \ldots, L_1}_{d}, \ldots, \underbrace{L_n, \ldots, L_n}_{d}) \in \mathbb{R}^{nd\times nd}$, and slightly abuse the notation to use $\boldsymbol{\Lambda}_\pi = \operatorname{diag}(\underbrace{L_{\pi^1}, \ldots, L_{\pi^1}}_{d}, \ldots, \underbrace{L_{\pi^n}, \ldots, L_{\pi^n}}_{d})$ given a permutation π of $[n]$. For the permutation π_k at the k-th epoch, we denote $\boldsymbol{\Lambda}_k = \boldsymbol{\Lambda}_{\pi_k}$, for brevity.

We further assume that the variance at $\boldsymbol{x}_*$ is bounded, same as prior work [MKR20, NTDP+21].

Assumption 2. *The quantity $\sigma_*^2 = \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_*)\|^2$ is bounded.*

2.3.1 Primal-dual view of shuffled SGD

Problem (P) can be reformulated into a primal-dual form using the standard Fenchel conjugacy argument (see, e.g., [CP11, CERS18]),

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \max_{\boldsymbol{y}\in\mathbb{R}^{nd}} \Big\{ \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) := \frac{1}{n}\sum_{i=1}^n \Big(\Big\langle \boldsymbol{y}^i, \boldsymbol{x} \Big\rangle - f_i^*(\boldsymbol{y}^i) \Big) = \frac{1}{n}\langle \boldsymbol{E}\boldsymbol{x}, \boldsymbol{y}\rangle - \frac{1}{n}\sum_{i=1}^n f_i^*(\boldsymbol{y}^i) \Big\}, \tag{PD}$$

where we slightly abuse the notation in this section and use $\boldsymbol{y}^i \in \mathbb{R}^d$ to be the i-th d elements of the vector $\boldsymbol{y}$ such that $\boldsymbol{y} = (\boldsymbol{y}^1, \dots, \boldsymbol{y}^n)^\top \in \mathbb{R}^{nd}$, $\boldsymbol{E} = [\underbrace{\boldsymbol{I}_d, \dots, \boldsymbol{I}_d}_{n}]^\top \in \mathbb{R}^{nd \times d}$ is the vertical concatenation of n identity matrices $\boldsymbol{I}_d \in \mathbb{R}^{d\times d}$ and f_i^* is the convex conjugate of f_i defined by $f_i(\boldsymbol{x}) = \sup_{\boldsymbol{y}^i \in \mathbb{R}^d} \langle \boldsymbol{y}^i, \boldsymbol{x}\rangle - f_i^*(\boldsymbol{y}^i)$. In the following, we consider the mini-batch estimator of batch size b, and let $\boldsymbol{y}^{(i)} \in \mathbb{R}^{bd}$ denote the vector comprised of the i^{th} bd elements of $\boldsymbol{y}$. For simplicity and without loss of generality, we assume that $n = bm$ for some positive integer m, so that $\boldsymbol{y} = (\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(m)})^\top$. Note that if choosing $b = 1$, our setting is the same as the ones in [MKR20, NTDP+21]. Then we have the primal-dual view of shuffled SGD scheme for general smooth convex minimization as in Alg. 1, where $\boldsymbol{E}_b^\top = [\underbrace{\boldsymbol{I}_d, \dots, \boldsymbol{I}_d}_{b}]^\top \in \mathbb{R}^{bd\times d}$ is the vertical concatenation of b identity matrices $\boldsymbol{I}_d \in \mathbb{R}^{d\times d}$. Given the data permutation $\pi^{(k)} = \{\pi_1^{(k)}, \pi_2^{(k)}, \dots, \pi_n^{(k)}\}$ of $[n]$ at the k-th epoch, we use the same notation of $\boldsymbol{v}_k = (\boldsymbol{v}^{\pi_1^{(k)}}, \dots, \boldsymbol{v}^{\pi_n^{(k)}})^\top \in \mathbb{R}^{nd}$, $\boldsymbol{y}_{*,k} = (\boldsymbol{y}_*^{\pi_1^{(k)}}, \dots, \boldsymbol{y}_*^{\pi_n^{(k)}})^\top \in \mathbb{R}^{nd}$ as in previous sections except now each $\boldsymbol{v}^{\pi_i^{(k)}}, \boldsymbol{y}_*^{\pi_i^{(k)}}$ are d-dimensional subvectors. Further, we denote the permuted smoothness constant matrices by $\boldsymbol{\Lambda}_k = \operatorname{diag}(\underbrace{L_{\pi_1^{(k)}}, \dots, L_{\pi_1^{(k)}}}_{d}, \dots, \underbrace{L_{\pi_n^{(k)}}, \dots, L_{\pi_n^{(k)}}}_{d}) \in \mathbb{R}^{nd\times nd}$, and we use $\boldsymbol{I}$ for $\boldsymbol{I}_{nd} \in \mathbb{R}^{nd\times nd}$ throughout this section.

Given a primal-dual pair $(\boldsymbol{x}, \boldsymbol{y})$, the primal-dual gap of (PD) is defined by

$$\operatorname{Gap}(\boldsymbol{x}, \boldsymbol{y}) = \max_{(\boldsymbol{u},\boldsymbol{v})} \{\mathcal{L}(\boldsymbol{x}, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \boldsymbol{y})\}.$$

In particular, we consider the pair $(\boldsymbol{x}, \boldsymbol{y}_*)$ for $\boldsymbol{x} \in \mathbb{R}^d$, and bound $\operatorname{Gap}^{\boldsymbol{v}}(\boldsymbol{x}, \boldsymbol{y}_*) := \mathcal{L}(\boldsymbol{x}, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*)$ for an arbitrary but fixed $\boldsymbol{v}$. To finally obtain the function value gap $f(\boldsymbol{x}) - f(\boldsymbol{x}_*)$ for (P), we only need to choose $\boldsymbol{v} = \arg\max_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{y}_{\boldsymbol{x}}$.

Using this primal-dual formulation and standard convex conjugacy arguments, we can *equivalently* write the standard shuffled SGD algorithm in a primal-dual form as summarized in Algorithm 1.

Improved bounds with new smoothness constants. To simplify the notation in the following lemmas and to clearly compare our results, we introduce the following novel definitions of smoothness constants for shuffled SGD

$$\begin{aligned} \hat{L}_\pi^g &:= \frac{1}{mn} \|\boldsymbol{\Lambda}_\pi^{1/2} (\textstyle\sum_{i=1}^m \boldsymbol{I}_{bd(i-1)\uparrow} \boldsymbol{E}\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow}) \boldsymbol{\Lambda}_\pi^{1/2}\|_2, \qquad && \hat{L}^g = \max_\pi \hat{L}_\pi^g, \\ \tilde{L}_\pi^g &:= \frac{1}{b} \|\boldsymbol{\Lambda}_\pi^{1/2} (\textstyle\sum_{i=1}^m \boldsymbol{I}_{(di)} \boldsymbol{E}\boldsymbol{E}^\top \boldsymbol{I}_{(di)}) \boldsymbol{\Lambda}_\pi^{1/2}\|_2, && \tilde{L}^g = \max_\pi \tilde{L}_\pi^g, \end{aligned} \tag{2.1}$$

where $\boldsymbol{I}_{(di)} = \sum_{j=bd(i-1)+1}^{bdi} \boldsymbol{I}_j$. Permutation-dependent quantities $\hat{L}_\pi^g$ and $\tilde{L}_\pi^g$ defined in (2.1) are obtained directly from our analysis. We remark that $\hat{L}^g$ is bounded by the average smoothness of f and $\tilde{L}^g$ is bounded by the max of individual smoothness constants of f_i. However, as we argue in

Algorithm 1 Shuffled SGD (Primal-Dual View, General Convex Smooth)

Input: Initial point $\boldsymbol{x}_0 \in \mathbb{R}^d$, step size $\{\eta_k\} > 0$, number of epochs $K > 0$
for $k = 1$ to K **do**
  Generate some permutation $\pi^{(k)}$ of $[n]$ (either deterministic or random)
  $\boldsymbol{x}_{k-1,1} = \boldsymbol{x}_{k-1}$
  for $i = 1$ to n in the ordering of $\pi^{(k)}$ **do**
    $\boldsymbol{y}_k^i = \arg\max_{\boldsymbol{y}^i \in \mathbb{R}^d} \left\{ \langle \boldsymbol{y}^i, \boldsymbol{x}_{k-1,i} \rangle - f_i^*(\boldsymbol{y}^i) \right\}$
    $\boldsymbol{x}_{k-1,i+1} = \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} \left\{ \langle \boldsymbol{y}_k^i, \boldsymbol{x} \rangle + \frac{1}{2\eta_k} \|\boldsymbol{x} - \boldsymbol{x}_{k-1,i}\|^2 \right\} = \boldsymbol{x}_{k-1,i} - \eta_k \nabla f_i(\boldsymbol{x}_{k-1,i})$
  end for
  $\boldsymbol{x}_k = \boldsymbol{x}_{k-1,n+1}$
end for
Return: $\hat{\boldsymbol{x}}_K = \sum_{k=1}^K \eta_k \boldsymbol{x}_k / \sum_{k=1}^K \eta_k$

later sections, these upper bounds on $\hat{L}_\pi^g$ and $\tilde{L}_\pi^g$ are loose in general, and so the convergence bounds based on $\hat{L}_\pi^g$ and $\tilde{L}_\pi^g$ that we obtain align better with the empirical performance of shuffled SGD.

To compare $\hat{L}_\pi^g$ and $L := \max_{i \in [n]} L_i$, we make use of the Kronecker product with notation $\otimes$ defined by

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} A_{11}\boldsymbol{B} & \cdots & A_{1n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ A_{m1}\boldsymbol{B} & \cdots & A_{nn}\boldsymbol{B} \end{bmatrix}$$

for two matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{p \times q}$. The following lemma states a useful fact for the Kronecker product.

Lemma 1. *For square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of sizes p and q and with eigenvalues λ_i ($i \in [p]$) and μ_j ($j \in [q]$) respectively, the eigenvalues of $\boldsymbol{A} \otimes \boldsymbol{B}$ are $\lambda_i \mu_j$ for $i \in [p], j \in [q]$.*

We now use the following chain of inequalities to compare $\hat{L}_\pi^g$ and L for any permutation π of $[n]$:

$$\begin{aligned} \hat{L}_\pi^g &= \frac{1}{mn} \Big\| \boldsymbol{\Lambda}_\pi^{1/2} \Big(\sum_{i=1}^m \boldsymbol{I}_{bd(i-1)\uparrow} \boldsymbol{E}\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow} \Big) \boldsymbol{\Lambda}_\pi^{1/2} \Big\|_2 \\ &\le \frac{1}{n} \Big\| \boldsymbol{\Lambda}^{1/2} \boldsymbol{E}\boldsymbol{E}^\top \boldsymbol{\Lambda}^{1/2} \Big\|_2 \\ &= \frac{1}{n} \Big\| (\boldsymbol{l}_\pi \boldsymbol{l}_\pi^\top) \otimes \boldsymbol{I}_d \Big\|_2 \\ &\overset{(i)}{=} \frac{1}{n} \sum_{i=1}^n L_i \le L, \end{aligned}$$

where we define $\boldsymbol{l}_\pi = (\sqrt{L_{\pi_1}}, \sqrt{L_{\pi_2}}, \ldots, \sqrt{L_{\pi_n}})^\top$. For (i), we use Lemma 1 and notice that the eigenvalues of $\boldsymbol{I}_d$ all equal 1, while the largest eigenvalue of $\boldsymbol{l}_\pi \boldsymbol{l}_\pi^\top = \|\boldsymbol{l}\|_2^2 = \sum_{i=1}^n L_i$, so the operator norm of $(\boldsymbol{l}_k \boldsymbol{l}_k^\top) \otimes \boldsymbol{I}_d$ is $\sum_{i=1}^n L_i$.

To compare $\tilde{L}_\pi^g$ and L, we notice that

$$\boldsymbol{\Lambda}_\pi^{1/2}(\textstyle\sum_{i=1}^m \boldsymbol{I}_{(di)} \boldsymbol{E}\boldsymbol{E}^\top \boldsymbol{I}_{(di)})\boldsymbol{\Lambda}_\pi^{1/2} = \sum_{i=1}^m \boldsymbol{I}_{(di)} \boldsymbol{\Lambda}_\pi^{1/2} \boldsymbol{E}\boldsymbol{E}^\top \boldsymbol{\Lambda}_\pi^{1/2} \boldsymbol{I}_{(di)}$$

is a block diagonal matrix whose operator norm is the maximum of the operator norms over its diagonal block submatrices, so we have

$$\begin{aligned}
\tilde{L}_\pi^g &= \frac{1}{b} \max_{i\in[m]} \left\| \boldsymbol{I}_{(di)} \boldsymbol{\Lambda}_\pi^{1/2} \boldsymbol{E}\boldsymbol{E}^\top \boldsymbol{\Lambda}_\pi^{1/2} \boldsymbol{I}_{(di)} \right\| \\
&= \frac{1}{b} \max_{i\in[m]} \left\| \boldsymbol{I}_{(di)} ((\boldsymbol{l}_\pi \boldsymbol{l}_\pi^\top) \otimes \boldsymbol{I}_d) \boldsymbol{I}_{(di)} \right\| \\
&\overset{(i)}{=} \max_{i\in[m]} \frac{1}{b} \sum_{j=1}^b L_{\pi_{b(i-1)+j}} \leq L,
\end{aligned}$$

where for (i) we use Lemma 1 for each submatrix $(\boldsymbol{l}_\pi^{(i)} \boldsymbol{l}_\pi^{(i)\top}) \otimes \boldsymbol{I}_d$ and

$$\boldsymbol{l}_\pi^{(i)} = (0, \ldots, 0, \sqrt{L_{\pi_{b(i-1)+1}}}, \ldots, \sqrt{L_{\pi_{bi}}}, 0, \ldots, 0)^\top.$$

Similar to the case of generalized linear models, the inequality is tight when $b = 1$ but can be loose for other values of b.

Before proceeding to the main proofs, we first state the following standard definitions and first-order characterization of strong convexity, for completeness.

Definition 6. *A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be μ-strongly convex with parameter $\mu > 0$, if for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and any $\lambda \in (0, 1)$:*

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}) - \frac{\mu}{2} \lambda (1-\lambda) \|\boldsymbol{x} - \boldsymbol{y}\|_2^2.$$

Lemma 2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuous μ-strongly convex function with $\mu > 0$. Then, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$:*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{g}_{\boldsymbol{x}}, \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\mu}{2} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2,$$

where $\boldsymbol{g}_{\boldsymbol{x}} \in \partial f(\boldsymbol{x})$, and $\partial f(\boldsymbol{x})$ is the subdifferential of f at $\boldsymbol{x}$.

We also include the following lemma on the variance bound under without-replacement sampling, which is useful for our proof.

Lemma 3. *Let $\mathcal{B}$ be the set of $|\mathcal{B}|= b$ samples from $[n]$, drawn without replacement and uniformly at random. Then, $\forall \boldsymbol{x} \in \mathbb{R}^d$,*

$$\mathbb{E}_{\mathcal{B}}\Big[\|\frac{1}{b}\sum_{i\in\mathcal{B}}\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|_2^2\Big] = \frac{n-b}{b(n-1)}\mathbb{E}_i[\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|_2^2].$$

Proof. We first expand the square on the left-hand side, as follows

$$\begin{aligned}
&\mathbb{E}_{\mathcal{B}}\Big[\|\frac{1}{b}\sum_{i\in\mathcal{B}}\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|_2^2\Big] \\
&= \frac{1}{b^2}\mathbb{E}_{\mathcal{B}}\Big[\sum_{i,i'\in\mathcal{B}}\langle\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x}), \nabla f_{i'}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\rangle\Big] \\
&= \frac{1}{b^2}\mathbb{E}_{\mathcal{B}}\Big[\sum_{i,i'\in\mathcal{B},i\neq i'}\langle\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x}), \nabla f_{i'}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\rangle\Big] + \frac{1}{b}\mathbb{E}_i[\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|_2^2].
\end{aligned}$$

Since the batch $\mathcal{B}$ is sampled uniformly and without replacement from $[n]$, the probability that any pair (i, i') from $[n]$ with $i \neq i'$ is in $\mathcal{B}$ is $\frac{b(b-1)}{n(n-1)}$. By the linearity of expectation, we have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{B}}\Big[\sum_{i,i'\in\mathcal{B},i\neq i'}\langle\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x}), \nabla f_{i'}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\rangle\Big] \\
&= \mathbb{E}_{\mathcal{B}}\Big[\sum_{i,i'\in[n],i\neq i'}\mathbb{1}_{i,i'\in\mathcal{B}}\langle\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x}), \nabla f_{i'}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\rangle\Big] \\
&= \sum_{i,i'\in[n],i\neq i'}\mathbb{E}_{\mathcal{B}}\Big[\mathbb{1}_{i,i'\in\mathcal{B}}\langle\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x}), \nabla f_{i'}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\rangle\Big] \\
&= \frac{b(b-1)}{n(n-1)}\sum_{i,i'\in[n],i\neq i'}\langle\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x}), \nabla f_{i'}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\rangle,
\end{aligned}$$

where $\mathbb{1}$ is the indicator function such that $\mathbb{1}_{i,i'\in\mathcal{B}} = 1$ if both $i, i' \in \mathcal{B}$ and is equal to zero otherwise. Hence, we obtain

$$\begin{aligned}
&\mathbb{E}_{\mathcal{B}}\Big[\|\frac{1}{b}\sum_{i\in\mathcal{B}}\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2\Big] \\
&= \frac{b-1}{bn(n-1)}\sum_{i,i'\in[n],i\neq i'}\langle\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x}), \nabla f_{i'}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\rangle + \frac{1}{b}\mathbb{E}_i[\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2] \\
&= \frac{b-1}{bn(n-1)}\sum_{i,i'\in[n]}\langle\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x}), \nabla f_{i'}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\rangle + \frac{n-b}{b(n-1)}\mathbb{E}_i[\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2] \\
&\overset{(i)}{=} \frac{n-b}{b(n-1)}\mathbb{E}_i[\|\nabla^j f_i(\boldsymbol{x}) - \nabla^j f(\boldsymbol{x})\|^2],
\end{aligned}$$

where (i) is due to $f = \frac{1}{n}\sum_{i=1}^n f_i$ having the finite sum structure. □

Now we provide the main proofs for obtaining a tighter fine-grained convergence bound for shuffled SGD in the general mini-batch setting.

Lemma 4. *Under Assumption 1, for any $k \in [K]$, the iterates $\{\boldsymbol{y}_k^{(i)}\}_{i=1}^m$ and $\{\boldsymbol{x}_{k-1,i}\}_{i=1}^{m+1}$ generated by Algorithm 1 satisfy*

$$\begin{aligned}\mathcal{E}_k \leq \frac{\eta_k}{n}\sum_{i=1}^m \Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_k^{(i)}, \boldsymbol{x}_k - \boldsymbol{x}_{k-1,i+1}\Big\rangle + \frac{\eta_k}{n}\sum_{i=1}^m \Big\langle \boldsymbol{E}_b^\top(\boldsymbol{v}_k^{(i)} - \boldsymbol{y}_k^{(i)}), \boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}\Big\rangle \\ - \frac{\eta_k}{2n}\|\boldsymbol{y}_k - \boldsymbol{v}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}} - \frac{\eta_k}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}} - \frac{b}{2n}\sum_{i=1}^m \|\boldsymbol{x}_{k-1,i+1} - \boldsymbol{x}_{k-1,i}\|^2,\end{aligned} \tag{2.2}$$

where $\mathcal{E}_k := \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*) + \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_k\|_2^2 - \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1}\|_2^2$.

Proof. We first note that based on Line 6 of Alg. 1, we have

$$\Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}^{(i)}, \boldsymbol{x}_{k-1,i}\Big\rangle - \sum_{j=1}^b f^*_{\pi^{(k)}_{b(i-1)+j}}(\boldsymbol{y}^j) = \sum_{j=1}^b \Big(\Big\langle \boldsymbol{y}^j, \boldsymbol{x}_{k-1,i}\Big\rangle - f^*_{\pi^{(k)}_{b(i-1)+j}}(\boldsymbol{y}^j)\Big).$$

Since the max problem defining $\boldsymbol{y}_k$ is separable, we have for $b(i-1)+1 \leq j \leq bi$ and $i \in [m]$

$$\boldsymbol{y}_k^j = \arg\max_{\boldsymbol{y}^j \in \mathbb{R}^d}\Big\{\Big\langle \boldsymbol{y}^j, \boldsymbol{x}_{k-1,i}\Big\rangle - f^*_{\pi_j^{(k)}}(\boldsymbol{y}^j)\Big\},$$

which leads to $\boldsymbol{x}_{k-1,i} \in \partial f^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j)$. Further, since each component function f_j^* is $\frac{1}{L_j}$-strongly convex thus for $b(i-1)+1 \leq j \leq bi$, we also have

$$f^*_{\pi_j^{(k)}}(\boldsymbol{v}_k^j) \geq f^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j) + \Big\langle \boldsymbol{x}_{k-1,i}, \boldsymbol{v}_k^j - \boldsymbol{y}_k^j\Big\rangle + \frac{1}{2L_{\pi_j^{(k)}}}\|\boldsymbol{v}_k^j - \boldsymbol{y}_k^j\|^2,$$

which leads to

$$\begin{aligned}&\mathcal{L}(\boldsymbol{x}_k, \boldsymbol{v}) \\ &= \frac{1}{n}\sum_{i=1}^m \Big(\Big\langle \boldsymbol{E}_b^\top \boldsymbol{v}_k^{(i)}, \boldsymbol{x}_{k-1,i}\Big\rangle - \sum_{j=b(i-1)+1}^{bi} f^*_{\pi_j^{(k)}}(\boldsymbol{v}_k^j)\Big) + \frac{1}{n}\sum_{i=1}^m \Big\langle \boldsymbol{E}_b^\top \boldsymbol{v}_k^{(i)}, \boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}\Big\rangle \\ &\leq \frac{1}{n}\sum_{i=1}^m \Big(\Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_k^{(i)}, \boldsymbol{x}_{k-1,i}\Big\rangle - \sum_{j=b(i-1)+1}^{bi} f^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j)\Big) + \frac{1}{n}\sum_{i=1}^m \Big\langle \boldsymbol{E}_b^\top \boldsymbol{v}_k^{(i)}, \boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}\Big\rangle \\ &\quad - \frac{1}{2n}\|\boldsymbol{y}_k - \boldsymbol{v}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}}.\end{aligned}$$

Using the same argument, as $\boldsymbol{x}_* \in \partial f_i^*(\boldsymbol{y}_*^i)$ for $i \in [n]$, we have

$$f^*_{\pi_i^{(k)}}(\boldsymbol{y}_k^i) \geq f^*_{\pi_i^{(k)}}(\boldsymbol{y}_{*,k}^i) + \Big\langle \boldsymbol{x}_*, \boldsymbol{y}_k^i - \boldsymbol{y}_{*,k}^i\Big\rangle + \frac{1}{2L_{\pi_i^{(k)}}}\|\boldsymbol{y}_k^i - \boldsymbol{y}_{*,k}^i\|^2.$$

Thus,

$$
\begin{aligned}
&\mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*) \\
&= \frac{1}{n}\sum_{i=1}^{m}\Big(\Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_{*,k}^{(i)}, \boldsymbol{x}_*\Big\rangle - \sum_{j=b(i-1)+1}^{bi} f^*_{\pi_j^{(k)}}(\boldsymbol{y}_{*,k}^j)\Big) \\
&\geq \frac{1}{n}\sum_{i=1}^{m}\Big(\Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_{k}^{(i)}, \boldsymbol{x}_*\Big\rangle - \sum_{j=b(i-1)+1}^{bi} f^*_{\pi_j^{(k)}}(\boldsymbol{y}_{k}^j)\Big) + \frac{1}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}} \\
&= \frac{1}{n}\sum_{i=1}^{m}\Big(\Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_{k}^{(i)}, \boldsymbol{x}_*\Big\rangle + \frac{b}{2\eta_k}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|^2 - \frac{b}{2\eta_k}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|^2 - \sum_{j=b(i-1)+1}^{bi} f^*_{\pi_j^{(k)}}(\boldsymbol{y}_{k}^j)\Big) \\
&\quad + \frac{1}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}}.
\end{aligned}
$$

Using the updating scheme of $\boldsymbol{x}_{k-1,i+1}$ and noticing that $\phi_k^i(\boldsymbol{x}) = \Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_k^{(i)}, \boldsymbol{x}\Big\rangle + \frac{b}{2\eta_k}\|\boldsymbol{x} - \boldsymbol{x}_{k-1,i}\|^2$ is $\frac{b}{\eta_k}$-strongly convex and minimized at $\boldsymbol{x}_{k-1,i+1}$, we have

$$
\begin{aligned}
&\Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_k^{(i)}, \boldsymbol{x}_*\Big\rangle + \frac{b}{2\eta_k}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|^2 \\
\geq\ &\Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_k^{(i)}, \boldsymbol{x}_{k-1,i+1}\Big\rangle + \frac{b}{2\eta_k}\|\boldsymbol{x}_{k-1,i+1} - \boldsymbol{x}_{k-1,i}\|^2 + \frac{b}{2\eta_k}\|\boldsymbol{x}_{k-1,i+1} - \boldsymbol{x}_*\|^2,
\end{aligned}
$$

which leads to

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*) &\geq \frac{1}{n}\sum_{i=1}^{m}\Big(\Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_k^{(i)}, \boldsymbol{x}_{k-1,i+1}\Big\rangle + \frac{b}{2\eta_k}\|\boldsymbol{x}_{k-1,i+1} - \boldsymbol{x}_{k-1,i}\|^2 - \sum_{j=b(i-1)+1}^{bi} f^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^i)\Big) \\
&\quad + \frac{b}{2n\eta_k}\sum_{i=1}^{m}\Big(\|\boldsymbol{x}_{k-1,i+1} - \boldsymbol{x}_*\|^2 - \|\boldsymbol{x}_{k-1,i} - \boldsymbol{x}_*\|^2\Big) + \frac{1}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}} \\
&= \frac{1}{n}\sum_{i=1}^{m}\Big(\Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_k^{(i)}, \boldsymbol{x}_{k-1,i+1}\Big\rangle + \frac{b}{2\eta_k}\|\boldsymbol{x}_{k-1,i+1} - \boldsymbol{x}_{k-1,i}\|^2 - \sum_{j=b(i-1)+1}^{bi} f^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j)\Big) \\
&\quad + \frac{b}{2n\eta_k}\Big(\|\boldsymbol{x}_k - \boldsymbol{x}_*\|^2 - \|\boldsymbol{x}_{k-1} - \boldsymbol{x}_*\|^2\Big) + \frac{1}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}}.
\end{aligned}
$$

Hence, combining the bounds on $\mathcal{L}(\boldsymbol{x}_k, \boldsymbol{v})$ and $\mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*)$ and letting

$$
\mathcal{E}_k := \eta_k(\mathcal{L}(\boldsymbol{x}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*)) + \frac{b}{2n}\|\boldsymbol{x}_k - \boldsymbol{x}_*\|^2 - \frac{b}{2n}\|\boldsymbol{x}_{k-1} - \boldsymbol{x}_*\|^2,
$$

we obtain

$$\begin{aligned}\mathcal{E}_k &\le \frac{\eta_k}{n}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_k^{(i)}, \boldsymbol{x}_{k-1,i}-\boldsymbol{x}_{k-1,i+1}\Big\rangle + \frac{\eta_k}{n}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}_b^\top \boldsymbol{v}_k^{(i)}, \boldsymbol{x}_k-\boldsymbol{x}_{k-1,i}\Big\rangle \\ &\quad - \frac{\eta_k}{2n}\|\boldsymbol{y}_k-\boldsymbol{v}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}} - \frac{\eta_k}{2n}\|\boldsymbol{y}_k-\boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}} - \frac{b}{2n}\sum_{i=1}^{m}\|\boldsymbol{x}_{k-1,i+1}-\boldsymbol{x}_{k-1,i}\|^2 \\ &= \frac{\eta_k}{n}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_k^{(i)}, \boldsymbol{x}_k-\boldsymbol{x}_{k-1,i+1}\Big\rangle + \frac{\eta_k}{n}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}_b^\top (\boldsymbol{v}_k^{(i)}-\boldsymbol{y}_k^{(i)}), \boldsymbol{x}_k-\boldsymbol{x}_{k-1,i}\Big\rangle \\ &\quad - \frac{\eta_k}{2n}\|\boldsymbol{y}_k-\boldsymbol{v}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}} - \frac{\eta_k}{2n}\|\boldsymbol{y}_k-\boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}} - \frac{b}{2n}\sum_{i=1}^{m}\|\boldsymbol{x}_{k-1,i+1}-\boldsymbol{x}_{k-1,i}\|^2,\end{aligned}$$

thus completing the proof. □

We note that the first inner product term $\mathcal{T}_1 := \frac{\eta_k}{n}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}_b^\top \boldsymbol{y}_k^{(i)}, \boldsymbol{x}_k-\boldsymbol{x}_{k-1,i+1}\Big\rangle$ in Eq. (2.2) can be cancelled by the last negative term $-\frac{b}{2n}\sum_{i=1}^{m}\|\boldsymbol{x}_{k-1,i+1}-\boldsymbol{x}_{k-1,i}\|^2$ therein, as precisely proved in Lemma 8 of Section 2.4. In the following subsections, we continue our analysis and handle the remaining terms in Eq. (2.2) according to different shuffling and derive the final complexity.

2.3.2 Random reshuffling/shuffle-once schemes

We introduce the following lemma to bound the second inner product term

$$\mathcal{T}_2 := \frac{\eta_k}{n}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}_b^\top (\boldsymbol{v}_k^{(i)}-\boldsymbol{y}_k^{(i)}), \boldsymbol{x}_k-\boldsymbol{x}_{k-1,i}\Big\rangle$$

in Lemma 4 when there are random permutations.

Lemma 5. *Under Assumptions 1 and 2, for any $k \in [K]$, the iterates $\{\boldsymbol{y}_k^{(i)}\}_{i=1}^m$ and $\{\boldsymbol{x}_{k-1,i}\}_{i=1}^{m+1}$ generated by Algorithm 1 with uniformly random shuffling (RR/SO) satisfy*

$$\mathbb{E}[\mathcal{T}_2] \le \mathbb{E}\Big[\frac{\eta_k^3 n \hat{L}^g_{\pi^{(k)}}\tilde{L}^g_{\pi^{(k)}}}{b^2}\|\boldsymbol{y}_k-\boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}} + \frac{\eta_k}{2n}\|\boldsymbol{v}_k-\boldsymbol{y}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}}\Big] + \frac{\eta_k^3\tilde{L}^g(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2,$$

where $\mathcal{T}_2 := \frac{\eta_k}{n}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}_b^\top (\boldsymbol{v}_k^{(i)}-\boldsymbol{y}_k^{(i)}), \boldsymbol{x}_k-\boldsymbol{x}_{k-1,i}\Big\rangle$.

Proof. First note that

$$\boldsymbol{x}_k-\boldsymbol{x}_{k-1,i} = \sum_{j=i}^{m}(\boldsymbol{x}_{k-1,j+1}-\boldsymbol{x}_{k-1,j}) = -\frac{\eta_k}{b}\sum_{j=i}^{m}\boldsymbol{E}_b^\top \boldsymbol{y}_k^{(j)} = -\frac{\eta_k}{b}\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_k,$$

so we have

$$
\begin{aligned}
&\frac{\eta_k}{n}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}_b^\top(\boldsymbol{v}_k^{(i)}-\boldsymbol{y}_k^{(i)}),\boldsymbol{x}_k-\boldsymbol{x}_{k-1,i}\Big\rangle\\
=&\frac{\eta_k}{n}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}_b^\top(\boldsymbol{v}_k^{(i)}-\boldsymbol{y}_k^{(i)}),\sum_{j=i}^{m}(\boldsymbol{x}_{k-1,j+1}-\boldsymbol{x}_{k-1,j})\Big\rangle\\
=&-\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}^\top\boldsymbol{I}_{(di)}(\boldsymbol{v}_k-\boldsymbol{y}_k),\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_k\Big\rangle\\
=&\underbrace{-\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}^\top\boldsymbol{I}_{(di)}(\boldsymbol{v}_k-\boldsymbol{y}_k),\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}(\boldsymbol{y}_k-\boldsymbol{y}_{*,k})\Big\rangle}_{\mathcal{I}_1}\\
&\underbrace{-\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}^\top\boldsymbol{I}_{(di)}(\boldsymbol{v}_k-\boldsymbol{y}_k),\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_{*,k}\Big\rangle}_{\mathcal{I}_2}.
\end{aligned}
$$

For the term $\mathcal{I}_1$, we use Young's inequality with $\alpha>0$ to be set later and obtain

$$
\begin{aligned}
\mathcal{I}_1=&-\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}^\top\boldsymbol{I}_{(di)}(\boldsymbol{v}_k-\boldsymbol{y}_k),\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}(\boldsymbol{y}_k-\boldsymbol{y}_{*,k})\Big\rangle\\
\le&\frac{\eta_k^2}{2bn\alpha}\sum_{i=1}^{m}\|\boldsymbol{E}^\top\boldsymbol{I}_{(di)}(\boldsymbol{v}_k-\boldsymbol{y}_k)\|^2+\frac{\eta_k^2\alpha}{2bn}\sum_{i=1}^{m}\|\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}(\boldsymbol{y}_k-\boldsymbol{y}_{*,k})\|^2.
\end{aligned}\tag{2.3}
$$

Further, notice that

$$
\begin{aligned}
&\frac{\eta_k^2\alpha}{2bn}\sum_{i=1}^{m}\|\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}(\boldsymbol{y}_k-\boldsymbol{y}_{*,k})\|^2\\
=&\frac{\eta_k^2\alpha}{2bn}\sum_{i=1}^{m}(\boldsymbol{y}_k-\boldsymbol{y}_{*,k})^\top\boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{E}\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}(\boldsymbol{y}_k-\boldsymbol{y}_{*,k})\\
=&\frac{\eta_k^2\alpha}{2bn}(\boldsymbol{y}_k-\boldsymbol{y}_{*,k})^\top\Big(\sum_{i=1}^{m}\boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{E}\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}\Big)(\boldsymbol{y}_k-\boldsymbol{y}_{*,k})\\
=&\frac{\eta_k^2\alpha}{2bn}(\boldsymbol{y}_k-\boldsymbol{y}_{*,k})^\top\boldsymbol{\Lambda}_k^{-1/2}\boldsymbol{\Lambda}_k^{1/2}\Big(\sum_{i=1}^{m}\boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{E}\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}\Big)\boldsymbol{\Lambda}_k^{1/2}\boldsymbol{\Lambda}_k^{-1/2}(\boldsymbol{y}_k-\boldsymbol{y}_{*,k})\\
\le&\frac{\eta_k^2\alpha}{2bn}\Big\|\boldsymbol{\Lambda}_k^{1/2}\Big(\sum_{i=1}^{m}\boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{E}\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}\Big)\boldsymbol{\Lambda}_k^{1/2}\Big\|_2\|\boldsymbol{y}_k-\boldsymbol{y}_{*,k}\|_{\boldsymbol{\Lambda}_k^{-1}}^2\\
=&\frac{\eta_k^2 m\alpha}{2b}\hat{L}_{\pi^{(k)}}^{g}\|\boldsymbol{y}_k-\boldsymbol{y}_{*,k}\|_{\boldsymbol{\Lambda}_k^{-1}}^2,
\end{aligned}\tag{2.4}
$$

where for the last inequality we use Cauchy-Schwarz inequality. Using the same argument, we can bound

$$\frac{\eta_k^2}{2bn\alpha}\sum_{i=1}^{m}\|\boldsymbol{E}^\top \boldsymbol{I}_{(di)}(\boldsymbol{v}_k-\boldsymbol{y}_k)\|^2 \le \frac{\eta_k^2}{2bn\alpha}\Big\|\boldsymbol{\Lambda}_k^{1/2}\Big(\sum_{i=1}^{m}\boldsymbol{I}_{(di)}\boldsymbol{E}\boldsymbol{E}^\top \boldsymbol{I}_{(di)}\Big)\boldsymbol{\Lambda}_k^{1/2}\Big\|_2\|\boldsymbol{v}_k-\boldsymbol{y}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}}$$
$$= \frac{\eta_k^2}{2n\alpha}\tilde{L}^g_{\pi^{(k)}}\|\boldsymbol{v}_k-\boldsymbol{y}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}}. \tag{2.5}$$

Thus, combining (2.3)–(2.5) and choosing $\alpha = 2\eta_k\tilde{L}^g_{\pi^{(k)}}$, we obtain

$$\mathcal{I}_1 \le \frac{\eta_k^3 m\hat{L}^g_{\pi^{(k)}}\tilde{L}^g_{\pi^{(k)}}}{b}\|\boldsymbol{y}_k-\boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}}+\frac{\eta_k}{4n}\|\boldsymbol{v}_k-\boldsymbol{y}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}}.$$

For the term $\mathcal{I}_2$, we again apply Young's inequality with $\beta>0$ to be set later and obtain

$$\begin{aligned}\mathcal{I}_2 =& -\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{E}^\top \boldsymbol{I}_{(di)}(\boldsymbol{v}_k-\boldsymbol{y}_k), \boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_{*,k}\Big\rangle\\ \le& \frac{\eta_k^2\beta}{2bn}\sum_{i=1}^{m}\|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_{*,k}\|^2+\frac{\eta_k^2}{2bn\beta}\sum_{i=1}^{m}\|\boldsymbol{E}^\top \boldsymbol{I}_{(di)}(\boldsymbol{v}_k-\boldsymbol{y}_k)\|^2\\ \le& \frac{\eta_k^2\beta}{2bn}\sum_{i=1}^{m}\|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_{*,k}\|^2+\frac{\eta_k^2\tilde{L}^g_{\pi^{(k)}}}{2n\beta}\|\boldsymbol{v}_k-\boldsymbol{y}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}}.\end{aligned}$$

Choosing $\beta = 2\eta_k\tilde{L}^g$ and using the fact that $\tilde{L}^g_{\pi^{(k)}}\le\tilde{L}^g$, we have

$$\mathcal{I}_2 \le \frac{\eta_k^3\tilde{L}^g}{bn}\sum_{i=1}^{m}\|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_{*,k}\|^2+\frac{\eta_k}{4n}\|\boldsymbol{v}_k-\boldsymbol{y}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}}.$$

Hence, combining the above two estimates with $m=n/b$, we have

$$\mathcal{T}_2 \le \frac{\eta_k^3\tilde{L}^g}{bn}\sum_{i=1}^{m}\|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_{*,k}\|^2+\frac{\eta_k^3 n\hat{L}^g_{\pi^{(k)}}\tilde{L}^g_{\pi^{(k)}}}{b^2}\|\boldsymbol{y}_k-\boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}}+\frac{\eta_k}{2n}\|\boldsymbol{v}_k-\boldsymbol{y}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}}.$$

First, consider the RR scheme. Taking conditional expectation on both sides w.r.t. the randomness up to but not including k-th epoch, we have

$$\begin{aligned}\mathbb{E}_k[\mathcal{T}_2] \le& \frac{\eta_k^3\tilde{L}^g}{bn}\mathbb{E}_k\Big[\sum_{i=1}^{m}\|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_{*,k}\|^2\Big]\\ &+\mathbb{E}_k\Big[\frac{\eta_k^3 n\hat{L}^g_{\pi^{(k)}}\tilde{L}^g_{\pi^{(k)}}}{b^2}\|\boldsymbol{y}_k-\boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}}+\frac{\eta_k}{2n}\|\boldsymbol{v}_k-\boldsymbol{y}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}}\Big].\end{aligned}$$

For the first term, since the only randomness comes from the permutation $\pi^{(k)}$, we can proceed as in the proof of Lemma 9 and obtain

$$\begin{aligned}\frac{\eta_k^3\tilde{L}^g}{bn}\mathbb{E}_k\Big[\sum_{i=1}^m\|\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_{*,k}\|^2\Big] &\overset{(i)}{=} \frac{\eta_k^3\tilde{L}^g}{bn}\sum_{i=1}^m\mathbb{E}_{\pi^{(k)}}\Big[\|\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_{*,k}\|^2\Big]\\ &= \frac{\eta_k^3\tilde{L}^g}{bn}\sum_{i=1}^m(n-b(i-1))^2\mathbb{E}_{\pi^{(k)}}\Big[\Big\|\frac{\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_{*,k}}{n-b(i-1)}\Big\|^2\Big]\\ &\overset{(ii)}{\leq} \frac{\eta_k^3\tilde{L}^g}{bn}\sum_{i=1}^m(n-b(i-1))^2\frac{b(i-1)}{(n-b(i-1))(n-1)}\sigma_*^2\\ &= \frac{\eta_k^3\tilde{L}^g(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2,\end{aligned}$$

where we use the linearity of expectation for (i), and (ii) is due to Lemma 3 and the definition $\sigma_*^2 = \frac{1}{n}\sum_{i=1}^n\|\nabla f_i(\boldsymbol{x}_*)\|^2 = \frac{1}{n}\sum_{i=1}^n\|\boldsymbol{y}_*^i\|^2$. Then taking expectation w.r.t. all randomness on both sides, we obtain

$$\mathbb{E}[\mathcal{T}_2] \leq \mathbb{E}\Big[\frac{\eta_k^3 n\hat{L}^g_{\pi^{(k)}}\tilde{L}^g_{\pi^{(k)}}}{b^2}\|\boldsymbol{y}_k-\boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}}+\frac{\eta_k}{2n}\|\boldsymbol{v}_k-\boldsymbol{y}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}}\Big]+\frac{\eta_k^3\tilde{L}^g(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

Finally, we remark that the above argument for bounding the term $\frac{\eta_k^3\tilde{L}^g}{bn}\mathbb{E}_k\Big[\sum_{i=1}^m\|\boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_{*,k}\|^2\Big]$ also applies to the SO scheme, in which case there is only one random permutation at the very beginning that induces the randomness. □

We state the final convergence rate and complexity in the following theorem and provide the proof for completeness.

Theorem 1. *Under Assumptions 1 and 2, if* $\eta_k \leq \frac{b}{n\sqrt{2\hat{L}^g_{\pi^{(k)}}\tilde{L}^g_{\pi^{(k)}}}}$ *and* $H_K = \sum_{k=1}^K\eta_k$*, the output* $\hat{\boldsymbol{x}}_K$ *of Algorithm 1 with uniformly random (RR/SO) shuffling satisfies*

$$\mathbb{E}[H_K(f(\hat{\boldsymbol{x}}_K)-f(\boldsymbol{x}_*))] \leq \frac{b}{2n}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2+\sum_{k=1}^K\frac{\eta_k^3\tilde{L}^g(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

As a consequence, for any $\epsilon>0$*, there exists a choice of a constant step size* $\eta_k=\eta$ *for which* $\mathbb{E}[f(\hat{\boldsymbol{x}}_K)-f(\boldsymbol{x}_*)]\leq\epsilon$ *after* $\mathcal{O}(\frac{n\sqrt{\hat{L}^g\tilde{L}^g}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{\epsilon}+\sqrt{\frac{(n-b)(n+b)}{n(n-1)}}\frac{\sqrt{n\tilde{L}^g}\sigma_*\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{\epsilon^{3/2}})$ *gradient queries.*

Proof. Combining the bounds in Lemma 4 and 5 and plugging them into Eq. (2.2), we obtain

$$\mathbb{E}[\mathcal{E}_k] \leq \mathbb{E}\Big[\Big(\frac{\eta_k^3 n\hat{L}^g_{\pi^{(k)}}\tilde{L}^g_{\pi^{(k)}}}{b^2}-\frac{\eta_k}{2n}\Big)\|\boldsymbol{y}_k-\boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}}\Big]+\frac{\eta_k^3\tilde{L}^g(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

For the stepsize η_k such that $\eta_k \le \frac{b}{n\sqrt{2\hat{L}^g_{\pi(k)}\tilde{L}^g_{\pi(k)}}}$, we have $\frac{\eta_k^3 n\hat{L}^g_{\pi(k)}\tilde{L}^g_{\pi(k)}}{b^2} - \frac{\eta_k}{2n} \le 0$, thus

$$\mathbb{E}[\mathcal{E}_k] \le \frac{\eta_k^3\tilde{L}^g(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

Using our definition of $\mathcal{E}_k$ and telescoping from $k=1$ to K, we have

$$\mathbb{E}\Big[\sum_{k=1}^K \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*)\Big] \le \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_0\|_2^2 - \frac{b}{2n}\mathbb{E}[\|\boldsymbol{x}_* - \boldsymbol{x}_K\|_2^2] + \sum_{k=1}^K \frac{\eta_k^3\tilde{L}^g(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

Noticing that $\boldsymbol{\mathcal{L}}(\boldsymbol{x}, \boldsymbol{v})$ is convex in $\boldsymbol{x}$ for a fixed $\boldsymbol{v}$, we have $\mathrm{Gap}^{\boldsymbol{v}}(\hat{\boldsymbol{x}}_K, \boldsymbol{y}_*) \le \sum_{k=1}^K \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*)/H_K$, where $\hat{\boldsymbol{x}}_K = \sum_{k=1}^K \eta_k \boldsymbol{x}_k / H_K$ and $H_K = \sum_{k=1}^K \eta_k$, which leads to

$$\mathbb{E}\Big[H_K \mathrm{Gap}^{\boldsymbol{v}}(\hat{\boldsymbol{x}}_K, \boldsymbol{y}_*)\Big] \le \frac{b}{2n}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^K \frac{\eta_k^3\tilde{L}^g(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

Further choosing $\boldsymbol{v} = \boldsymbol{y}_{\hat{\boldsymbol{x}}_K}$, we obtain

$$\mathbb{E}[H_K(f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*))] \le \frac{b}{2n}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^K \frac{\eta_k^3\tilde{L}^g(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2. \tag{2.6}$$

To analyze the individual gradient oracle complexity, we choose constant stepsizes $\eta \le \frac{b}{n\sqrt{2\hat{L}^g\tilde{L}^g}}$, then Eq. (2.6) will become

$$\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \le \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{\eta^2\tilde{L}^g(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

Without loss of generality, we assume that $b \ne n$, otherwise the method and its analysis reduce to (full) gradient descent. We consider the following two cases:

- "Small K" case: if $\eta = \frac{b}{n\sqrt{2\hat{L}^g\tilde{L}^g}} \le \Big(\frac{3b^3(n-1)\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{n(n-b)(n+b)\tilde{L}^g K\sigma_*^2}\Big)^{1/3}$, we have

$$\begin{aligned}
&\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \\
&\le \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{\eta^2\tilde{L}^g(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2 \\
&\le \frac{\sqrt{\hat{L}^g\tilde{L}^g}}{\sqrt{2}K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{1}{2}\Big(\frac{(n-b)(n+b)}{n^2(n-1)}\Big)^{1/3}\frac{(\tilde{L}^g)^{1/3}\sigma_*^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{3^{1/3}K^{2/3}}.
\end{aligned}$$

- "Large K" case: if $\eta = \Big(\frac{3b^3(n-1)\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{n(n-b)(n+b)\tilde{L}^g K\sigma_*^2}\Big)^{1/3} \le \frac{b}{n\sqrt{2\hat{L}^g\tilde{L}^g}}$, we have

$$\begin{aligned}
\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] &\le \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{\eta^2\tilde{L}^g(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2 \\
&\le \Big(\frac{(n-b)(n+b)}{n^2(n-1)}\Big)^{1/3}\frac{(\tilde{L}^g)^{1/3}\sigma_*^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{3^{1/3}K^{2/3}}.
\end{aligned}$$

Combining these two cases by setting $\eta = \min\Big\{\frac{b}{n\sqrt{2\hat{L}^g\tilde{L}^g}}, \Big(\frac{3b^3(n-1)\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{n(n-b)(n+b)\tilde{L}^gK\sigma_*^2}\Big)^{1/3}\Big\}$, we obtain

$$\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \leq \frac{\sqrt{\hat{L}^g\tilde{L}^g}}{\sqrt{2}K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \Big(\frac{(n-b)(n+b)}{n^2(n-1)}\Big)^{1/3}\frac{(\tilde{L}^g)^{1/3}\sigma_*^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{3^{1/3}K^{2/3}}.$$

Hence, to guarantee $\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \leq \epsilon$ for $\epsilon > 0$, the total number of individual gradient evaluations will be

$$nK \geq \max\Big\{\frac{n\sqrt{2\hat{L}^g\tilde{L}^g}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon}, \Big(\frac{(n-b)(n+b)}{n-1}\Big)^{1/2}\frac{2^{3/2}(\tilde{L}^g)^{1/2}\sigma_*\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{3^{1/2}\epsilon^{3/2}}\Big\},$$

as claimed. □

2.3.3 Incremental gradient descent (IG)

In this subsection, we provide the convergence results for incremental gradient descent which does not involve random permutations. We first prove the technical lemma below to bound the term $\mathcal{T}_2 := \frac{\eta_k}{n}\sum_{i=1}^m \Big\langle \boldsymbol{E}_b^\top(\boldsymbol{v}_k^{(i)} - \boldsymbol{y}_k^{(i)}), \boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}\Big\rangle$ in Eq. (2.2) of Lemma 4.

Lemma 6. *For any $k \in [K]$, the iterates $\{\boldsymbol{y}_k^{(i)}\}_{i=1}^m$ and $\{\boldsymbol{x}_{k-1,i}\}_{i=1}^{m+1}$ generated by Algorithm 1 with fixed data ordering satisfy*

$$\begin{aligned}\mathcal{T}_2 \leq{}& \frac{\eta_k^3 n}{b^2}\hat{L}_0^g\tilde{L}_0^g\|\boldsymbol{y}_k - \boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 + \frac{\eta_k}{2n}\|\boldsymbol{v} - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}^{-1}}^2 \\ &+ \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0^g\tilde{L}_0^g\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0^g\sigma_*^2\Big\}.\end{aligned} \tag{2.7}$$

Proof. Proceeding as in the proof of Lemma 5, we have

$$\begin{aligned}&\frac{\eta_k}{n}\sum_{i=1}^m \Big\langle \boldsymbol{E}_b^\top(\boldsymbol{v}^{(i)} - \boldsymbol{y}_k^{(i)}), \boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}\Big\rangle \\ ={}& \frac{\eta_k}{n}\sum_{i=1}^m \Big\langle \boldsymbol{E}_b^\top(\boldsymbol{v}^{(i)} - \boldsymbol{y}_k^{(i)}), \sum_{j=i}^m(\boldsymbol{x}_{k-1,j+1} - \boldsymbol{x}_{k-1,j})\Big\rangle \\ ={}& -\frac{\eta_k^2}{bn}\sum_{i=1}^m \Big\langle \boldsymbol{E}^\top\boldsymbol{I}_{(di)}(\boldsymbol{v} - \boldsymbol{y}_k), \boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_k\Big\rangle \\ ={}& \underbrace{-\frac{\eta_k^2}{bn}\sum_{i=1}^m \Big\langle \boldsymbol{E}^\top\boldsymbol{I}_{(di)}(\boldsymbol{v} - \boldsymbol{y}_k), \boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}(\boldsymbol{y}_k - \boldsymbol{y}_*)\Big\rangle}_{\mathcal{I}_1} \\ &\underbrace{-\frac{\eta_k^2}{bn}\sum_{i=1}^m \Big\langle \boldsymbol{E}^\top\boldsymbol{I}_{(di)}(\boldsymbol{v} - \boldsymbol{y}_k), \boldsymbol{E}^\top\boldsymbol{I}_{bd(i-1)\uparrow}\boldsymbol{y}_*\Big\rangle}_{\mathcal{I}_2}.\end{aligned}$$

For both terms $\mathcal{I}_1$ and $\mathcal{I}_2$, we apply Young's inequality with $\alpha = 2\eta_k \tilde{L}_0^g$ and obtain

$$\begin{aligned}
\mathcal{I}_1 &\leq \frac{\eta_k^2 \alpha}{2bn} \sum_{i=1}^{m} \|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow}(\boldsymbol{y}_k - \boldsymbol{y}_*)\|_2^2 + \frac{\eta_k^2}{2bn\alpha} \sum_{i=1}^{m} \|\boldsymbol{E}^\top \boldsymbol{I}_{(di)}(\boldsymbol{v} - \boldsymbol{y}_k)\|_2^2 \\
&\leq \frac{\eta_k^2 n\alpha}{2b^2} \hat{L}_0^g \|\boldsymbol{y}_k - \boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 + \frac{\eta_k^2}{2n\alpha} \tilde{L}_0^g \|\boldsymbol{v} - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}^{-1}}^2 \\
&= \frac{\eta_k^3 n}{b^2} \hat{L}_0^g \tilde{L}_0^g \|\boldsymbol{y}_k - \boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 + \frac{\eta_k}{4n} \|\boldsymbol{v} - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}^{-1}}^2,
\end{aligned} \tag{2.8}$$

and

$$\begin{aligned}
\mathcal{I}_2 &\leq \frac{\eta_k^2 \alpha}{2bn} \sum_{i=1}^{m} \|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow} \boldsymbol{y}_*\|_2^2 + \frac{\eta_k^2}{2bn\alpha} \sum_{i=1}^{m} \|\boldsymbol{E}^\top \boldsymbol{I}_{(di)}(\boldsymbol{v} - \boldsymbol{y}_k)\|_2^2 \\
&\leq \frac{\eta_k^2 \alpha}{2bn} \sum_{i=1}^{m} \|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow} \boldsymbol{y}_*\|_2^2 + \frac{\eta_k^2}{2n\alpha} \tilde{L}_0^g \|\boldsymbol{v} - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}^{-1}}^2 \\
&= \frac{\eta_k^3 \tilde{L}_0^g}{nb} \sum_{i=1}^{m} \|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow} \boldsymbol{y}_*\|_2^2 + \frac{\eta_k}{4n} \|\boldsymbol{v} - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}^{-1}}^2.
\end{aligned} \tag{2.9}$$

We now show that the term $\frac{\eta_k^3 \tilde{L}_0^g}{nb} \sum_{i=1}^{m} \|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow} \boldsymbol{y}_*\|_2^2$ is no larger than either $\frac{\eta_k^3 n}{b^2} \hat{L}_0^g \tilde{L}_0^g \|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2$ or $\frac{\eta_k^3 (n-b)^2}{b^2} \tilde{L}_0^g \sigma_*^2$. This is trivial when $b = n$ as $\boldsymbol{E}^\top \boldsymbol{I}_{0\uparrow} \boldsymbol{y}_* = \sum_{i=1}^{n} \boldsymbol{y}_*^i = \boldsymbol{0}$. When $b < n$, to show the former one, we have

$$\begin{aligned}
\sum_{i=1}^{m} \|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow} \boldsymbol{y}_*\|_2^2 &\leq \Big\| \boldsymbol{\Lambda}^{1/2} \Big(\sum_{i=1}^{m} \boldsymbol{I}_{bd(i-1)\uparrow} \boldsymbol{E} \boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow} \Big) \boldsymbol{\Lambda}^{1/2} \Big\|_2 \|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 \\
&= mn \hat{L}_0^g \|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 = \frac{n^2}{b} \hat{L}_0^g \|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2.
\end{aligned}$$

To prove the latter one, we notice that

$$\begin{aligned}
\sum_{i=1}^{m} \|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow} \boldsymbol{y}_*\|_2^2 &= \sum_{i=1}^{m} \Big\| \sum_{j=b(i-1)+1}^{n} \boldsymbol{y}_*^j \Big\|_2^2 = \sum_{i=0}^{m-1} \Big\| \sum_{j=bi+1}^{n} \boldsymbol{y}_*^j \Big\|_2^2 = \sum_{i=1}^{m-1} \Big\| \sum_{j=bi+1}^{n} \boldsymbol{y}_*^j \Big\|_2^2 \\
&= \sum_{i=1}^{m-1} \Big\| \sum_{j=1}^{bi} \boldsymbol{y}_*^j \Big\|_2^2,
\end{aligned}$$

using the fact that $\sum_{i=1}^{n} \boldsymbol{y}_*^i = \mathbf{0}$. Then using Young's inequality we obtain

$$\begin{aligned}\sum_{i=1}^{m-1} \Big\| \sum_{j=1}^{bi} \boldsymbol{y}_*^j \Big\|_2^2 &\leq \sum_{i=1}^{m-1} bi \sum_{j=1}^{bi} \|\boldsymbol{y}_*^j\|_2^2 \\ &\leq b(m-1) \sum_{i=1}^{m-1} \sum_{j=1}^{bi} \|\boldsymbol{y}_*^j\|_2^2 \\ &= b(m-1) \sum_{i=1}^{m-1} \sum_{j=b(i-1)+1}^{bi} (m-i) \|\boldsymbol{y}_*^j\|_2^2 \\ &\leq b(m-1)^2 \sum_{i=1}^{(m-1)b} \|\boldsymbol{y}_*^i\|_2^2.\end{aligned}$$

Further noticing that $\sum_{i=1}^{(m-1)b} \|\boldsymbol{y}_*^i\|_2^2 \leq \sum_{i=1}^{n} \|\boldsymbol{y}_*^i\|^2 = n\sigma_*^2$, we have

$$\frac{\eta_k^3 \tilde{L}_0^g}{nb} \sum_{i=1}^{m} \|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow} \boldsymbol{y}_*\|_2^2 \leq \frac{\eta_k^3 \tilde{L}_0^g}{nb} b(m-1)^2 n\sigma_*^2 = \frac{\eta_k^3 \tilde{L}_0^g (n-b)^2}{b^2} \sigma_*^2.$$

The same bound also captures the case $b = n$ and leads to

$$\frac{\eta_k^3 \tilde{L}_0^g}{nb} \sum_{i=1}^{m} \|\boldsymbol{E}^\top \boldsymbol{I}_{bd(i-1)\uparrow} \boldsymbol{y}_*\|_2^2 \leq \min\Big\{ \frac{\eta_k^3 n}{b^2} \hat{L}_0^g \tilde{L}_0^g \|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3 (n-b)^2}{b^2} \tilde{L}_0^g \sigma_*^2 \Big\}. \tag{2.10}$$

Hence, combining Eq. (2.8)–(2.10), we obtain

$$\begin{aligned}\mathcal{I}_2 \leq{}& \frac{\eta_k^3 n}{b^2} \hat{L}_0^g \tilde{L}_0^g \|\boldsymbol{y}_k - \boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 + \frac{\eta_k}{2n} \|\boldsymbol{v} - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}^{-1}}^2 \\ &+ \min\Big\{ \frac{\eta_k^3 n}{b^2} \hat{L}_0^g \tilde{L}_0^g \|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3 (n-b)^2}{b^2} \tilde{L}_0^g \sigma_*^2 \Big\},\end{aligned}$$

which finishes the proof. □

We are now ready to state our convergence results for IGD in the following theorem, with its proof provided for completeness.

Theorem 2. *Under Assumptions 1 and 2, if* $\eta_k \leq \frac{b}{n\sqrt{2\hat{L}_0^g \tilde{L}_0^g}}$ *and* $H_K = \sum_{k=1}^{K} \eta_k$*, the output* $\hat{\boldsymbol{x}}_K$ *of Algorithm 1 with a fixed permutation satisfies*

$$H_K(f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)) \leq \frac{b}{2n} \|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^{K} \min\Big\{ \frac{\eta_k^3 n}{b^2} \hat{L}_0^g \tilde{L}_0^g \|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3 (n-b)^2}{b^2} \tilde{L}_0^g \sigma_*^2 \Big\}.$$

As a consequence, for any $\epsilon > 0$*, there exists a choice of a constant step size* $\eta_k = \eta$ *such that* $f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) \leq \epsilon$ *after* $\mathcal{O}\Big(\frac{n\sqrt{\hat{L}_0^g \tilde{L}_0^g} \|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon} + \frac{\min\big\{ \sqrt{n\hat{L}_0^g \tilde{L}_0^g} \|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}, (n-b)\sqrt{\tilde{L}_0^g} \sigma_* \big\} \|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon^{3/2}} \Big)$ *gradient queries.*

Proof. Combining the bounds in Lemma 8 and 6 and plugging them into Eq. (2.2) in Lemma 4 without random permutations, we have

$$\mathcal{E}_k \le \Big(\frac{\eta_k^3 n \hat{L}_0^g \tilde{L}_0^g}{b^2} - \frac{\eta_k}{2n}\Big)\|\boldsymbol{y}_k - \boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 + \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0^g\tilde{L}_0^g\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0^g\sigma_*^2\Big\}.$$

If $\eta_k \le \frac{b}{n\sqrt{2\hat{L}_0^g\tilde{L}_0^g}}$, we have $\frac{\eta_k^3 n\hat{L}_0^g\tilde{L}_0^g}{b^2} - \frac{\eta_k}{2n} \le 0$, thus

$$\mathcal{E}_k \le \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0^g\tilde{L}_0^g\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0^g\sigma_*^2\Big\}.$$

Using the definition of $\mathcal{E}_k$ and telescoping from $k=1$ to K, we obtain

$$\begin{aligned}\sum_{k=1}^K \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*) \le \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_0\|_2^2 - \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_K\|_2^2 \\ + \sum_{k=1}^K \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0^g\tilde{L}_0^g\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0^g\sigma_*^2\Big\}.\end{aligned}$$

Noticing that $\mathcal{L}(\boldsymbol{x}, \boldsymbol{v})$ is convex w.r.t. $\boldsymbol{x}$, we have $\mathrm{Gap}^{\boldsymbol{v}}(\hat{\boldsymbol{x}}_K, \boldsymbol{y}_*) \le \sum_{k=1}^K \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*)/H_K$, where $\hat{\boldsymbol{x}}_K = \sum_{k=1}^K \eta_k \boldsymbol{x}_k / H_K$ and $H_K = \sum_{k=1}^K \eta_k$, so we obtain

$$H_K \mathrm{Gap}^{\boldsymbol{v}}(\hat{\boldsymbol{x}}_K, \boldsymbol{y}_*) \le \frac{b}{2n}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^K \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0^g\tilde{L}_0^g\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0^g\sigma_*^2\Big\},$$

Further choosing $\boldsymbol{v} = \boldsymbol{y}_{\hat{\boldsymbol{x}}_K}$, we obtain

$$H_K(f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)) \le \frac{b}{2n}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^K \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0^g\tilde{L}_0^g\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0^g\sigma_*^2\Big\}. \tag{2.11}$$

To analyze the individual gradient oracle complexity, we choose constant stepsizes $\eta \le \frac{b}{n\sqrt{2\hat{L}_0^g\tilde{L}_0^g}}$ and assume $b < n$ without loss of generality, then Eq. (2.11) becomes

$$f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) \le \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \min\Big\{\frac{\eta^2 n}{b^2}\hat{L}_0^g\tilde{L}_0^g\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta^2(n-b)^2}{b^2}\tilde{L}_0^g\sigma_*^2\Big\}.$$

When $\hat{L}_0^g\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 \le \frac{(n-b)^2}{n}\sigma_*^2$, we set $\eta = \min\Big\{\frac{b}{n\sqrt{2\hat{L}_0^g\tilde{L}_0^g}}, \Big(\frac{b^3\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{2n^2\hat{L}_0^g\tilde{L}_0^g K\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2}\Big)^{1/3}\Big\}$ and consider the following two possible cases:

- "Small K" case: if $\eta = \frac{b}{n\sqrt{2\hat{L}_0^g\tilde{L}_0^g}} \le \Big(\frac{b^3\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{2n^2\hat{L}_0^g\tilde{L}_0^g K\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2}\Big)^{1/3}$, we have

$$\begin{aligned}f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) &\le \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{\eta^2 n}{b^2}\hat{L}_0^g\tilde{L}_0^g\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 \\ &\le \frac{\sqrt{\hat{L}_0^g\tilde{L}_0^g}}{\sqrt{2}K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{(\hat{L}_0^g\tilde{L}_0^g)^{1/3}\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{2^{2/3}n^{1/3}K^{2/3}}.\end{aligned}$$

- "Large K" case: if $\eta = \Big(\frac{b^3\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{2n^2\hat{L}_0^g\tilde{L}_0^gK\|\boldsymbol{y}_*\|^2_{\boldsymbol{\Lambda}^{-1}}}\Big)^{1/3} \leq \frac{b}{\sqrt{2\hat{L}_0^g\tilde{L}_0^g}}$, we have

$$\begin{aligned} f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) &\leq \frac{b}{2n\eta K}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2 + \frac{\eta^2 n}{b^2}\hat{L}_0^g\tilde{L}_0^g\|\boldsymbol{y}_*\|^2_{\boldsymbol{\Lambda}^{-1}} \\ &\leq \frac{2^{1/3}(\hat{L}_0^g\tilde{L}_0^g)^{1/3}\|\boldsymbol{y}_*\|^{2/3}_{\boldsymbol{\Lambda}^{-1}}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^{4/3}}{n^{1/3}K^{2/3}}. \end{aligned}$$

Combining these two cases, we have

$$f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) \leq \frac{\sqrt{\hat{L}_0^g\tilde{L}_0^g}}{\sqrt{2}K}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2 + \frac{2^{1/3}(\hat{L}_0^g\tilde{L}_0^g)^{1/3}\|\boldsymbol{y}_*\|^{2/3}_{\boldsymbol{\Lambda}^{-1}}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^{4/3}}{n^{1/3}K^{2/3}}.$$

Hence, to guarantee $\mathbb{E}[f(\hat{\boldsymbol{x}}_K)-f(\boldsymbol{x}_*)] \leq \epsilon$ for $\epsilon > 0$, the total number of required individual gradient evaluations will be

$$nK \geq \max\Big\{\frac{n\sqrt{2\hat{L}_0^g\tilde{L}_0^g}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{\epsilon}, \frac{4n^{1/2}(\hat{L}_0^g\tilde{L}_0^g)^{1/2}\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{\epsilon^{3/2}}\Big\}. \tag{2.12}$$

When $\frac{(n-b)^2}{n}\sigma_*^2 \leq \hat{L}_0^g\|\boldsymbol{y}_*\|^2_{\boldsymbol{\Lambda}^{-1}}$, we set $\eta = \min\Big\{\frac{b}{n\sqrt{2\hat{L}_0^g\tilde{L}_0^g}}, \Big(\frac{b^3\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{2n(n-b)^2\tilde{L}_0^gK\sigma_*^2}\Big)^{1/3}\Big\}$ and consider the two cases as below:

- "Small K" case: if $\eta = \frac{b}{n\sqrt{2\hat{L}_0^g\tilde{L}_0^g}} \leq \Big(\frac{b^3\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{2n(n-b)^2\tilde{L}_0^gK\sigma_*^2}\Big)^{1/3}$, we have

$$\begin{aligned} f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) &\leq \frac{b}{2n\eta K}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2 + \frac{\eta^2(n-b)^2}{b^2}\tilde{L}_0^g\sigma_*^2 \\ &\leq \frac{\sqrt{\hat{L}_0^g\tilde{L}_0^g}}{\sqrt{2}K}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2 + \frac{(n-b)^{2/3}(\tilde{L}^g)^{1/3}\sigma_*^{2/3}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^{4/3}}{2^{2/3}n^{2/3}K^{2/3}}. \end{aligned}$$

- "Large K" case: if $\eta = \Big(\frac{b^3\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{2n(n-b)^2\tilde{L}_0^gK\sigma_*^2}\Big)^{1/3} \leq \frac{b}{n\sqrt{2\hat{L}_0^g\tilde{L}_0^g}}$, we have

$$\begin{aligned} f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) &\leq \frac{b}{2n\eta K}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2 + \frac{\eta^2(n-b)^2}{b^2}\tilde{L}_0^g\sigma_*^2 \\ &\leq \frac{2^{1/3}(n-b)^{2/3}(\tilde{L}^g)^{1/3}\sigma_*^{2/3}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^{4/3}}{n^{2/3}K^{2/3}}. \end{aligned}$$

Combining these two cases, we obtain

$$f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) \leq \frac{\sqrt{\hat{L}_0^g\tilde{L}_0^g}}{\sqrt{2}K}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2 + \frac{2^{1/3}(n-b)^{2/3}(\tilde{L}^g)^{1/3}\sigma_*^{2/3}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^{4/3}}{n^{2/3}K^{2/3}}.$$

To guarantee $\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \leq \epsilon$ for $\epsilon > 0$, the total number of required individual gradient evaluations will be

$$nK \geq \max\Big\{\frac{n\sqrt{2\hat{L}_0^g\tilde{L}_0^g}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{\epsilon}, \frac{4(n-b)(\tilde{L}_0^g)^{1/2}\sigma_*\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{\epsilon^{3/2}}\Big\}. \tag{2.13}$$

Combining Eq. (2.12) and Eq. (2.13), we finally have

$$nK \geq \frac{n\sqrt{2\hat{L}_0^g \tilde{L}_0^g}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon} + \min\Big\{\frac{4n^{1/2}(\hat{L}_0^g \tilde{L}_0^g)^{1/2}\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon^{3/2}}, \frac{4(n-b)(\tilde{L}_0^g)^{1/2}\sigma_*\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon^{3/2}}\Big\},$$

thus finishing the proof. □

2.4 Tighter Bounds for Convex Finite-Sum Problems with Linear Predictors

To study the effect of the structure of the data on the convergence of shuffled SGD, we sharpen the focus from general finite-sum problems to convex finite-sum with linear predictors:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d}\Big\{f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^n \ell_i(\boldsymbol{a}_i^\top \boldsymbol{x})\Big\}, \tag{PL}$$

where $\boldsymbol{a}_i \in \mathbb{R}^d$ ($i \in [n]$) are data vectors and $\ell_i : \mathbb{R} \to \mathbb{R}$ are convex and either smooth or Lipschitz nonsmooth functions associated with the linear predictors $\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle$ for $i \in [n]$. In addition to their explicit dependence on the data, it is worth noting that problems of the form (PL) cover most of the standard convex ERM problems where shuffled SGD is commonly applied, such as support vector machines, least absolute deviation, least squares, and logistic regression.

Problem (PL) admits an explicit primal-dual formulation using the standard Fenchel conjugacy argument (see, e.g., [CP11, CERS18]),

$$\min_{\boldsymbol{x}\in\mathbb{R}^d}\max_{\boldsymbol{y}\in\mathbb{R}^n}\Big\{\mathcal{L}(\boldsymbol{x},\boldsymbol{y}) := \frac{1}{n}\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y}\rangle - \frac{1}{n}\sum_{i=1}^n \ell_i^*(\boldsymbol{y}^i) = \frac{1}{n}\sum_{i=1}^n (\boldsymbol{a}_i^\top \boldsymbol{x}\boldsymbol{y}^i - \ell_i^*(\boldsymbol{y}^i))\Big\}, \tag{PL-PD}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n]^\top \in \mathbb{R}^{n\times d}$ is the data matrix and $\ell_i^* : \mathbb{R} \to \mathbb{R}$ is the convex conjugate of ℓ_i. This observation allows us to again interpret without-replacement SGD updates as cyclic coordinate updates on the dual side. Note that due to the objective structure in (PL), the primal-dual formulation (PL-PD) can decouple the linear ($\boldsymbol{a}_i^\top \boldsymbol{x}$) and the non-linear ($\ell_i$) parts within individual loss functions f_i. We redefine the conjugate pair of $\boldsymbol{x} \in \mathbb{R}^d$ to be $\boldsymbol{y}_{\boldsymbol{x}} = (\boldsymbol{y}_{\boldsymbol{x}}^1, \ldots, \boldsymbol{y}_{\boldsymbol{x}}^n)^\top \in \mathbb{R}^n$, with $\boldsymbol{y}_{\boldsymbol{x}}^i = \arg\max_{\boldsymbol{y}^i\in\mathbb{R}}\{\boldsymbol{y}^i \boldsymbol{a}_i^\top \boldsymbol{x} - \ell_i^*(\boldsymbol{y}^i)\}$.

Based on the formulation (PL-PD), we view shuffled SGD as a primal-dual method with block coordinate updates on the dual side, as summarized in Algorithm 2, for completeness. To see the equivalence, in i-th inner iteration of k-th epoch, we first update the i-th block $\boldsymbol{y}_k^{(i)} \in \mathbb{R}^b$ of the dual vector $\boldsymbol{y}_{k-1} \in \mathbb{R}^n$ based on $\boldsymbol{x}_{k-1,i}$ as in Line 6. Since the dual update has a decomposable structure,

this maximization step corresponds to computing the (sub)gradients $\{\ell'_{\pi_j^{(k)}}(\boldsymbol{a}_{\pi_j^{(k)}}^\top \boldsymbol{x}_{k-1,i})\}_{j=b(i-1)+1}^{bi}$ at $\boldsymbol{x}_{k-1,i}$ for the batch of individual losses indexed by $\{\pi_j^{(k)}\}_{j=b(i-1)+1}^{bi}$. Then in Line 7, we perform a minimization step using $\boldsymbol{y}_k^{(i)}$ to compute $\boldsymbol{x}_{k-1,i+1}$ on the primal side. Combining these two steps, we have $\boldsymbol{x}_{k-1,i+1} = \boldsymbol{x}_{k-1,i} - \frac{\eta_k}{b}\sum_{j=b(i-1)+1}^{bi} \ell'_{\pi_j^{(k)}}(\boldsymbol{a}_{\pi_j^{(k)}}^\top \boldsymbol{x}_{k-1,i})\boldsymbol{a}_{\pi_j^{(k)}}$, which is exactly the *original primal shuffled SGD updating scheme*.

In this section, we consider shuffled SGD with *mini-batch* estimators of size b and assume without loss of generality that $n = bm$ for some positive integer m.

Algorithm 2 Shuffled SGD (Primal-Dual View)

1: **Input:** Initial point $\boldsymbol{x}_0 \in \mathbb{R}^d$, batch size $b > 0$, step size $\{\eta_k\} > 0$, number of epochs $K > 0$
2: **for** $k = 1$ to K **do**
3: Generate any permutation $\pi^{(k)}$ of $[n]$ (either deterministic or random)
4: $\boldsymbol{x}_{k-1,1} = \boldsymbol{x}_{k-1}$
5: **for** $i = 1$ to m **do**
6: $\boldsymbol{y}_k^{(i)} = \arg\max_{\boldsymbol{y}\in\mathbb{R}^b} \{\boldsymbol{y}^\top \boldsymbol{A}_k^{(i)} \boldsymbol{x}_{k-1,i} - \sum_{j=1}^b \ell^*_{\pi_{b(i-1)+j}^{(k)}}(\boldsymbol{y}^j)\}$
7: $\boldsymbol{x}_{k-1,i+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \{\boldsymbol{y}_k^{(i)\top} \boldsymbol{A}_k^{(i)} \boldsymbol{x} + \frac{b}{2\eta_k}\|\boldsymbol{x} - \boldsymbol{x}_{k-1,i}\|^2\}$
8: **end for**
9: $\boldsymbol{x}_k = \boldsymbol{x}_{k-1,m+1}$, $\boldsymbol{y}_k = (\boldsymbol{y}_k^{(1)}, \boldsymbol{y}_k^{(2)}, \ldots, \boldsymbol{y}_k^{(m)})^\top$
10: **end for**
11: **Return:** $\hat{\boldsymbol{x}}_K = \sum_{k=1}^K \eta_k \boldsymbol{x}_k / \sum_{k=1}^K \eta_k$

2.4.1 Smooth and convex objectives

Throughout this subsection, we make the following standard assumptions, corresponding to Assumptions 1 and 2 from Section 2.3.

Assumption 3. *Each ℓ_i is convex and L_i-smooth ($i \in [n]$), i.e., $|\ell_i'(x) - \ell_i'(y)| \le L_i|x - y|$ for any $x, y \in \mathbb{R}$. There exists a minimizer $\boldsymbol{x}_* \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$.*

We remark that Assumption 3 implies that both f and each component function $f_i(\boldsymbol{x}) = \ell_i(\boldsymbol{a}_i^\top \boldsymbol{x})$ are $L_{\max}$-smooth, where $L_{\max} = \max_{i\in[n]} L_i\|\boldsymbol{a}_i\|_2^2$. Assumption 3 also implies that each convex conjugate ℓ_i^* is $\frac{1}{L_i}$-strongly convex [Bec17]. In the following, we let $\boldsymbol{\Lambda} = \mathrm{diag}(L_1, L_2, \ldots, L_n)$, and $\boldsymbol{\Lambda}_\pi = \mathrm{diag}(L_{\pi^1}, L_{\pi^2}, \ldots, L_{\pi^n})$, given a permutation π of $[n]$.

We further assume bounded variance at $\boldsymbol{x}_*$, same as prior work [MKR20, NTDP$^+$21, TNTD21, TSN22].

Assumption 4. $\sigma_*^2 := \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_*)\|^2 = \frac{1}{n}\sum_{i=1}^n (\ell_i'(\boldsymbol{a}_i^\top \boldsymbol{x}_*))^2\|\boldsymbol{a}_i\|_2^2$ *is bounded.*

Improved bounds with new smoothness constants. Our convergence bounds depend on the smoothness parameters defined in Eq. (2.14) below. We provide a detailed discussion on how these parameters relate to traditional smoothness parameters both in the worst case and on typical datasets, in Section 2.5.1, with additional numerical results provided in Subsection 2.5.2.

$$\begin{aligned} \hat{L}_\pi &:= \frac{1}{mn}\|\boldsymbol{\Lambda}_\pi^{1/2}(\textstyle\sum_{j=1}^m \boldsymbol{I}_{b(j-1)\uparrow}\boldsymbol{A}_\pi\boldsymbol{A}_\pi^\top\boldsymbol{I}_{b(j-1)\uparrow})\boldsymbol{\Lambda}_\pi^{1/2}\|_2, & \hat{L} &= \max_\pi \hat{L}_\pi, \\ \tilde{L}_\pi &:= \frac{1}{b}\|\boldsymbol{\Lambda}_\pi^{1/2}(\textstyle\sum_{j=1}^m \boldsymbol{I}_{(j)}\boldsymbol{A}_\pi\boldsymbol{A}_\pi^\top\boldsymbol{I}_{(j)})\boldsymbol{\Lambda}_\pi^{1/2}\|_2, & \tilde{L} &= \max_\pi \tilde{L}_\pi, \end{aligned} \tag{2.14}$$

where $\boldsymbol{I}_{(j)} := \sum_{i=b(j-1)+1}^{bj} \boldsymbol{I}_i$. In comparison to the smoothness constants defined in Eq. (2.1) for general finite-sum problems, we note that the constants in Eq. (2.14) applying to generalized linear models are tighter and more informative estimates, as the data matrix $\boldsymbol{A}$ and the smoothness constants from the nonlinear part $\boldsymbol{\Lambda}$ are separated in Eq. (2.14). Thus, the constants $\hat{L}_\pi$ and $\tilde{L}_\pi$ directly depend on the data matrix, which explicitly demonstrates how the structure of the data affects the convergence of shuffled SGD.

2.4.2 Tighter Rates for Random Reshuffling/Shuffle-Once Schemes with Linear Predictors

Lemma 7. *Given $\{\boldsymbol{y}_k^{(i)}\}_{i=1}^m$ and $\{\boldsymbol{x}_{k-1,i}\}_{i=1}^{m+1}$ generated by Algorithm 2 for $k \in [K]$, let $\mathcal{E}_k := \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*) + \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_k\|_2^2 - \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1}\|_2^2$. If Assumption 3 holds, then*

$$\begin{aligned} \mathcal{E}_k \le{}& \frac{\eta_k}{n}\sum_{i=1}^m \boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i+1}) \\ &+ \frac{\eta_k}{n}\sum_{i=1}^m (\boldsymbol{v}_k^{(i)} - \boldsymbol{y}_k^{(i)})^\top \boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}) \\ &- \frac{\eta_k}{2n}\|\boldsymbol{y}_k - \boldsymbol{v}_k\|_{\boldsymbol{\Lambda}_k^{-1}}^2 - \frac{\eta_k}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|_{\boldsymbol{\Lambda}_k^{-1}}^2 \\ &- \frac{b}{2n}\sum_{i=1}^m \|\boldsymbol{x}_{k-1,i} - \boldsymbol{x}_{k-1,i+1}\|_2^2, \end{aligned} \tag{2.15}$$

Proof. By Line 6 in Alg. 2, we have $\boldsymbol{y}_k^{(i)} = \arg\max_{\boldsymbol{y}\in\mathbb{R}^b}\left\{\boldsymbol{y}^\top \boldsymbol{A}_k^{(i)}\boldsymbol{x}_{k-1,i} - \sum_{j=1}^b \ell^*_{\pi^{(k)}_{b(i-1)+j}}(\boldsymbol{y}^j)\right\}$ for $i \in [m]$. Notice that since

$$\boldsymbol{y}^\top \boldsymbol{A}_k^{(i)}\boldsymbol{x}_{k-1,i} - \sum_{j=1}^b \ell^*_{\pi^{(k)}_{b(i-1)+j}}(\boldsymbol{y}^j) = \sum_{j=1}^b \left(\boldsymbol{y}^j \boldsymbol{a}_{\pi^{(k)}_{b(i-1)+j}}^\top \boldsymbol{x}_{k-1,i} - \ell^*_{\pi^{(k)}_{b(i-1)+j}}(\boldsymbol{y}^j)\right)$$

is separable, we have $\boldsymbol{y}_k^j = \arg\max_{y\in\mathbb{R}}\{y\boldsymbol{a}_{\pi_j^{(k)}}^\top \boldsymbol{x}_{k-1,i} - \ell^*_{\pi_j^{(k)}}(y)\}$ for $b(i-1)+1 \le j \le bi$, thus $\boldsymbol{a}_{\pi_j^{(k)}}^\top \boldsymbol{x}_{k-1,i} \in \partial\ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j)$. Since ℓ_i^* is $\frac{1}{L_i}$-strongly convex by Assumption 3, then by Lemma 2 we

obtain for $b(i-1)+1 \le j \le bi$

$$\ell^*_{\pi_j^{(k)}}(\boldsymbol{v}_k^j) \ge \ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j) + \boldsymbol{a}^\top_{\pi_j^{(k)}}\boldsymbol{x}_{k-1,i}(\boldsymbol{v}_k^j - \boldsymbol{y}_k^j) + \frac{1}{2L_{\pi_j^{(k)}}}(\boldsymbol{v}_k^j - \boldsymbol{y}_k^j)^2,$$

which leads to

$$\begin{aligned}
\mathcal{L}(\boldsymbol{x}_k, \boldsymbol{v}) &= \frac{1}{n}\sum_{i=1}^m \Big(\boldsymbol{v}_k^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_{k-1,i} - \sum_{j=b(i-1)+1}^{bi} \ell^*_{\pi_j^{(k)}}(\boldsymbol{v}_k^j)\Big) + \frac{1}{n}\sum_{i=1}^m \boldsymbol{v}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}) \\
&\le \frac{1}{n}\sum_{i=1}^m \Big(\boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_{k-1,i} - \sum_{j=b(i-1)+1}^{bi} \ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j)\Big) + \frac{1}{n}\sum_{i=1}^m \boldsymbol{v}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}) \\
&\quad - \frac{1}{2n}\|\boldsymbol{y}_k - \boldsymbol{v}_k\|^2_{\boldsymbol{\Lambda}_k^{-1}}. \qquad (2.16)
\end{aligned}$$

Using the same argument for $\mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*)$ as $\boldsymbol{a}_j^\top \boldsymbol{x}_* \in \partial \ell_j^*(\boldsymbol{y}_*^j)$ for $j \in [n]$, we have

$$\begin{aligned}
\mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*) &= \frac{1}{n}\sum_{i=1}^m \Big(\boldsymbol{y}_{*,k}^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_* - \sum_{j=b(i-1)+1}^{bi} \ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_{*,k}^j)\Big) \\
&\ge \frac{1}{n}\sum_{i=1}^m \Big(\boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_* - \sum_{j=b(i-1)+1}^{bi} \ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j)\Big) + \frac{1}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}}. \qquad (2.17)
\end{aligned}$$

Adding and substracting the term $\frac{b}{2n\eta_k}\sum_{i=1}^m \|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|_2^2$ on the R.H.S. of Eq. (2.17), we obtain

$$\begin{aligned}
\mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*) \ge \frac{1}{n}\sum_{i=1}^m \Big(\boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_* + \frac{b}{2\eta_k}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|_2^2 - \sum_{j=b(i-1)+1}^{bi} \ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j)\Big) \\
- \frac{b}{2n\eta_k}\sum_{i=1}^m \|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|_2^2 + \frac{1}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}}.
\end{aligned}$$

By Line 7 of Alg. 2, we have $\boldsymbol{x}_{k-1,i+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d}\Big\{\boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x} + \frac{b}{2\eta_k}\|\boldsymbol{x} - \boldsymbol{x}_{k-1,i}\|_2^2\Big\}$. Further noticing that $\phi_k^{(i)}(\boldsymbol{x}) := \boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x} + \frac{b}{2\eta_k}\|\boldsymbol{x} - \boldsymbol{x}_{k-1,i}\|_2^2$ is $\frac{b}{\eta_k}$-strongly convex w.r.t. $\boldsymbol{x}$ and $\nabla\phi_k^{(i)}(\boldsymbol{x}_{k-1,i+1}) = \boldsymbol{0}$, we have

$$\phi_k^{(i)}(\boldsymbol{x}_*) \ge \phi_k^{(i)}(\boldsymbol{x}_{k-1,i+1}) + \frac{b}{2\eta_k}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i+1}\|_2^2,$$

which leads to

$$\begin{aligned}
\mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*) &\ge \frac{1}{n}\sum_{i=1}^m \Big(\boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_{k-1,i+1} + \frac{b}{2\eta_k}\|\boldsymbol{x}_{k-1,i+1} - \boldsymbol{x}_{k-1,i}\|_2^2 - \sum_{j=b(i-1)+1}^{bi} \ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j)\Big) \\
&\quad + \frac{b}{2n\eta_k}\sum_{i=1}^m (\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i+1}\|_2^2 - \|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|_2^2) + \frac{1}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}} \\
&\overset{(i)}{=} \frac{1}{n}\sum_{i=1}^m \Big(\boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_{k-1,i+1} + \frac{b}{2\eta_k}\|\boldsymbol{x}_{k-1,i+1} - \boldsymbol{x}_{k-1,i}\|_2^2 - \sum_{j=b(i-1)+1}^{bi} \ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j)\Big) \\
&\quad + \frac{b}{2n\eta_k}\|\boldsymbol{x}_k - \boldsymbol{x}_*\|_2^2 - \frac{b}{2n\eta_k}\|\boldsymbol{x}_{k-1} - \boldsymbol{x}_*\|_2^2 + \frac{1}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|^2_{\boldsymbol{\Lambda}_k^{-1}}, \qquad (2.18)
\end{aligned}$$

where we telescope from $i=1$ to m for the term $\sum_{i=1}^m (\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i+1}\|_2^2 - \|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|_2^2)$, and use the definitions that $\boldsymbol{x}_k = \boldsymbol{x}_{k-1,m+1}$ and $\boldsymbol{x}_{k-1} = \boldsymbol{x}_{k-1,1}$ for (i).

Combining the bounds from Eq. (2.16) and Eq. (2.18) and denoting

$$\mathcal{E}_k := \eta_k(\mathcal{L}(\boldsymbol{x}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*)) + \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_k\|_2^2 - \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1}\|_2^2,$$

we obtain

$$\begin{aligned}
\mathcal{E}_k &\le \frac{\eta_k}{n}\sum_{i=1}^m \boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_{k-1,i} - \boldsymbol{x}_{k-1,i+1}) + \frac{\eta_k}{n}\sum_{i=1}^m \boldsymbol{v}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}) \\
&\quad - \frac{\eta_k}{2n}\|\boldsymbol{y}_k - \boldsymbol{v}_k\|_{\boldsymbol{\Lambda}_k^{-1}}^2 - \frac{\eta_k}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|_{\boldsymbol{\Lambda}_k^{-1}}^2 - \frac{b}{2n}\sum_{i=1}^m \|\boldsymbol{x}_{k-1,i} - \boldsymbol{x}_{k-1,i+1}\|_2^2 \\
&= \frac{\eta_k}{n}\sum_{i=1}^m \boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i+1}) + \frac{\eta_k}{n}\sum_{i=1}^m (\boldsymbol{v}_k^{(i)} - \boldsymbol{y}_k^{(i)})^\top\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}) \\
&\quad - \frac{\eta_k}{2n}\|\boldsymbol{y}_k - \boldsymbol{v}_k\|_{\boldsymbol{\Lambda}_k^{-1}}^2 - \frac{\eta_k}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|_{\boldsymbol{\Lambda}_k^{-1}}^2 - \frac{b}{2n}\sum_{i=1}^m \|\boldsymbol{x}_{k-1,i} - \boldsymbol{x}_{k-1,i+1}\|_2^2,
\end{aligned}$$

thus completing the proof. □

Lemma 8. *For any $k \in [K]$, the iterates $\{\boldsymbol{y}_k^{(i)}\}_{i=1}^m$ and $\{\boldsymbol{x}_{k-1,i}\}_{i=1}^{m+1}$ in Algorithm 2 satisfy*

$$\mathcal{T}_1 = \frac{b}{2n}\sum_{i=1}^m \|\boldsymbol{x}_{k-1,i} - \boldsymbol{x}_{k-1,i+1}\|_2^2 - \frac{b}{2n}\|\boldsymbol{x}_{k-1} - \boldsymbol{x}_k\|_2^2.$$

Proof. By Line 7 in Alg. 2, we have $\boldsymbol{A}_k^{(i)\top}\boldsymbol{y}_k^{(i)} = \frac{b}{\eta_k}(\boldsymbol{x}_{k-1,i} - \boldsymbol{x}_{k-1,i+1})$. Further noticing that $\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i+1} = -\sum_{j=i+1}^m (\boldsymbol{x}_{k-1,j} - \boldsymbol{x}_{k-1,j+1})$, we obtain

$$\begin{aligned}
\mathcal{T}_1 &:= \frac{\eta_k}{n}\sum_{i=1}^m \boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i+1}) \\
&= -\frac{b}{n}\sum_{i=1}^{m-1}\sum_{j=i+1}^m \langle \boldsymbol{x}_{k-1,i} - \boldsymbol{x}_{k-1,i+1}, \boldsymbol{x}_{k-1,j} - \boldsymbol{x}_{k-1,j+1}\rangle \\
&= \frac{b}{2n}\sum_{i=1}^m \|\boldsymbol{x}_{k-1,i} - \boldsymbol{x}_{k-1,i+1}\|^2 - \frac{b}{2n}\Big\|\sum_{i=1}^m (\boldsymbol{x}_{k-1,i} - \boldsymbol{x}_{k-1,i+1})\Big\|^2,
\end{aligned}$$

thus completing the proof. □

Lemma 9. *Under Assumption 4, for any $k \in [K]$, the iterates $\{\boldsymbol{y}_k^{(i)}\}_{i=1}^m$ and $\{\boldsymbol{x}_{k-1,i}\}_{i=1}^{m+1}$ generated by Algorithm 2 with uniformly random shuffling satisfy*

$$\mathbb{E}[\mathcal{T}_2] \le \mathbb{E}\Big[\frac{\eta_k^3 n\hat{L}_{\pi^{(k)}}\tilde{L}_{\pi^{(k)}}}{b^2}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|_{\boldsymbol{\Lambda}_k^{-1}}^2 + \frac{\eta_k}{2n}\|\boldsymbol{v}_k - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}_k^{-1}}^2\Big] + \frac{\eta_k^3\tilde{L}(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

Proof. By Line 7 in Alg. 2, we have $\boldsymbol{x}_{k-1,i} - \boldsymbol{x}_{k-1,i+1} = \frac{\eta_k}{b}\boldsymbol{A}_k^{(i)\top}\boldsymbol{y}_k^{(i)}$. Using the definition of $\boldsymbol{I}_{j\uparrow}$ for $0 \le j \le n-1$ as in Section 2.1, we obtain

$$\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i} = -\sum_{j=i}^{m}(\boldsymbol{x}_{k-1,j} - \boldsymbol{x}_{k-1,j+1}) = -\frac{\eta_k}{b}\sum_{j=i}^{m}\boldsymbol{A}_k^{(j)\top}\boldsymbol{y}_k^{(j)} = -\frac{\eta_k}{b}\boldsymbol{A}_k\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_k.$$

Also, we have $\boldsymbol{A}_k^{(i)\top}(\boldsymbol{v}_k^{(i)} - \boldsymbol{y}_k^{(i)}) = \boldsymbol{A}_k\boldsymbol{I}_{(i)}(\boldsymbol{v}_k - \boldsymbol{y}_k)$ by the definition of $\boldsymbol{I}_{(i)}$ in Section 2.4. Combining these two observations, we have

$$\begin{aligned}
\mathcal{T}_2 :=& \frac{\eta_k}{n}\sum_{i=1}^{m}(\boldsymbol{v}_k^{(i)} - \boldsymbol{y}_k^{(i)})^\top\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}) \\
=& -\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_k, \boldsymbol{A}_k^\top\boldsymbol{I}_{(i)}(\boldsymbol{v}_k - \boldsymbol{y}_k)\Big\rangle \\
\overset{(i)}{=}& -\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}(\boldsymbol{y}_k - \boldsymbol{y}_{*,k}), \boldsymbol{A}_k^\top\boldsymbol{I}_{(i)}(\boldsymbol{v}_k - \boldsymbol{y}_k)\Big\rangle \qquad (2.19) \\
& -\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}, \boldsymbol{A}_k^\top\boldsymbol{I}_{(i)}(\boldsymbol{v}_k - \boldsymbol{y}_k)\Big\rangle, \qquad (2.20)
\end{aligned}$$

where we make a decomposition w.r.t. $\boldsymbol{y}_{*,k}$ in (i). For the first term in Eq. (2.19), we use Young's inequality for $\alpha > 0$ and have

$$\begin{aligned}
& -\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}(\boldsymbol{y}_k - \boldsymbol{y}_{*,k}), \boldsymbol{A}_k^\top\boldsymbol{I}_{(i)}(\boldsymbol{v}_k - \boldsymbol{y}_k)\Big\rangle \\
\le& \frac{\eta_k^2\alpha}{2bn}\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}(\boldsymbol{y}_k - \boldsymbol{y}_{*,k})\|_2^2 + \frac{\eta_k^2}{2bn\alpha}\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top\boldsymbol{I}_{(i)}(\boldsymbol{v}_k - \boldsymbol{y}_k)\|_2^2.
\end{aligned} \qquad (2.21)$$

Expanding the squares and rearranging the terms in Eq. (2.21), we have

$$\begin{aligned}
& \frac{\eta_k^2\alpha}{2bn}\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}(\boldsymbol{y}_k - \boldsymbol{y}_{*,k})\|_2^2 \\
=& \frac{\eta_k^2\alpha}{2bn}\sum_{i=1}^{m}(\boldsymbol{y}_k - \boldsymbol{y}_{*,k})^\top\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{A}_k\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}(\boldsymbol{y}_k - \boldsymbol{y}_{*,k}) \\
=& \frac{\eta_k^2\alpha}{2bn}(\boldsymbol{y}_k - \boldsymbol{y}_{*,k})^\top\Big(\sum_{i=1}^{m}\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{A}_k\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}\Big)(\boldsymbol{y}_k - \boldsymbol{y}_{*,k}) \qquad (2.22) \\
=& \frac{\eta_k^2\alpha}{2bn}(\boldsymbol{y}_k - \boldsymbol{y}_{*,k})^\top\boldsymbol{\Lambda}_k^{-1/2}\boldsymbol{\Lambda}_k^{1/2}\Big(\sum_{i=1}^{m}\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{A}_k\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}\Big)\boldsymbol{\Lambda}_k^{1/2}\boldsymbol{\Lambda}_k^{-1/2}(\boldsymbol{y}_k - \boldsymbol{y}_{*,k}) \\
\overset{(i)}{\le}& \frac{\eta_k^2\alpha}{2bn}\Big\|\boldsymbol{\Lambda}_k^{1/2}\Big(\sum_{i=1}^{m}\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{A}_k\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}\Big)\boldsymbol{\Lambda}_k^{1/2}\Big\|_2\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|_{\boldsymbol{\Lambda}_k^{-1}}^2,
\end{aligned}$$

where we use Cauchy-Schwarz inequality for (i). Using a similar argument, we also have

$$\frac{\eta_k^2}{2bn\alpha}\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top\boldsymbol{I}_{(i)}(\boldsymbol{v}_k - \boldsymbol{y}_k)\|_2^2 \le \frac{\eta_k^2}{2bn\alpha}\Big\|\boldsymbol{\Lambda}_k^{1/2}\Big(\sum_{i=1}^{m}\boldsymbol{I}_{(i)}\boldsymbol{A}_k\boldsymbol{A}_k^\top\boldsymbol{I}_{(i)}\Big)\boldsymbol{\Lambda}_k^{1/2}\Big\|_2\|\boldsymbol{v}_k - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}_k^{-1}}^2.$$

By the definitions of $\hat{L}_{\pi^{(k)}}$ and $\tilde{L}_{\pi^{(k)}}$, and choosing $\alpha = 2\eta_k \tilde{L}_{\pi^{(k)}}$ in Eq. (2.21), we obtain

$$\begin{aligned} &-\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{A}_k^\top \boldsymbol{I}_{b(i-1)\uparrow}(\boldsymbol{y}_k - \boldsymbol{y}_{*,k}), \boldsymbol{A}_k^\top \boldsymbol{I}_{(i)}(\boldsymbol{v}_k - \boldsymbol{y}_k)\Big\rangle \\ &\le \frac{\eta_k^3 n \hat{L}_{\pi^{(k)}} \tilde{L}_{\pi^{(k)}}}{b^2}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|_{\boldsymbol{\Lambda}_k^{-1}}^2 + \frac{\eta_k}{4n}\|\boldsymbol{v}_k - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}_k^{-1}}^2. \end{aligned} \tag{2.23}$$

For the second term in Eq. (2.20), we apply Young's inequality with $\beta > 0$ and proceed as above:

$$\begin{aligned} &-\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{A}_k^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}, \boldsymbol{A}_k^\top \boldsymbol{I}_{(i)}(\boldsymbol{v}_k - \boldsymbol{y}_k)\Big\rangle \\ &\le \frac{\eta_k^2\beta}{2bn}\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}\|_2^2 + \frac{\eta_k^2}{2bn\beta}\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top \boldsymbol{I}_{(i)}(\boldsymbol{v}_k - \boldsymbol{y}_k)\|_2^2 \\ &\le \frac{\eta_k^2\beta}{2bn}\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}\|_2^2 + \frac{\eta_k^2}{2n\beta}\tilde{L}_{\pi^{(k)}}\|\boldsymbol{v}_k - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}_k^{-1}}^2. \end{aligned}$$

Noticing that $\tilde{L}_{\pi^{(k)}} \le \tilde{L}$, we choose $\beta = 2\eta_k \tilde{L}$ and obtain

$$\begin{aligned} &-\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{A}_k^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}, \boldsymbol{A}_k^\top \boldsymbol{I}_{(i)}(\boldsymbol{v}_k - \boldsymbol{y}_k)\Big\rangle \\ &\le \frac{\eta_k^3 \tilde{L}}{nb}\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}\|_2^2 + \frac{\eta_k}{4n}\|\boldsymbol{v}_k - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}_k^{-1}}^2. \end{aligned} \tag{2.24}$$

Combining Eq. (2.23) and Eq. (2.24), we have

$$\mathcal{T}_2 \le \frac{\eta_k^3 \tilde{L}}{nb}\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}\|_2^2 + \frac{\eta_k^3 n \hat{L}_{\pi^{(k)}} \tilde{L}_{\pi^{(k)}}}{b^2}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|_{\boldsymbol{\Lambda}_k^{-1}}^2 + \frac{\eta_k}{2n}\|\boldsymbol{v}_k - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}_k^{-1}}^2. \tag{2.25}$$

We first assume the RR scheme. Taking conditional expectation w.r.t. the randomness up to but not including k-th epoch, we have

$$\begin{aligned} \mathbb{E}_k[\mathcal{T}_2] \le\ & \frac{\eta_k^3 \tilde{L}}{nb}\mathbb{E}_k\Big[\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}\|_2^2\Big] \\ &+ \mathbb{E}_k\Big[\frac{\eta_k^3 n \hat{L}_{\pi^{(k)}} \tilde{L}_{\pi^{(k)}}}{b^2}\|\boldsymbol{y}_k - \boldsymbol{y}_{*,k}\|_{\boldsymbol{\Lambda}_k^{-1}}^2 + \frac{\eta_k}{2n}\|\boldsymbol{v}_k - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}_k^{-1}}^2\Big]. \end{aligned}$$

For the first term $\frac{\eta_k^3 \tilde{L}}{nb}\mathbb{E}_k\Big[\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}\|_2^2\Big]$, the only randomness is from the random permutation $\pi^{(k)}$. In this case, each term $\boldsymbol{A}_k^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}$ can be considered as a sum of a batch sampled without replacement from $\{\boldsymbol{y}_*^j \boldsymbol{a}_j\}_{j\in[n]}$, while $\sum_{j=1}^{n}\boldsymbol{y}_*^j \boldsymbol{a}_j = 0$ as $\boldsymbol{x}_*$ is the minimizer, we then can use

Lemma 3 and obtain

$$\begin{aligned}\frac{\eta_k^3\tilde{L}}{nb}\mathbb{E}_k\Big[\sum_{i=1}^m\|\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}\|_2^2\Big] &\overset{(i)}{=} \frac{\eta_k^3\tilde{L}}{nb}\sum_{i=1}^m\mathbb{E}_{\pi^{(k)}}[\|\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}\|_2^2]\\ &= \frac{\eta_k^3\tilde{L}}{nb}\sum_{i=1}^m(n-b(i-1))^2\mathbb{E}_{\pi^{(k)}}\Big[\Big\|\frac{\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}}{n-b(i-1)}\Big\|_2^2\Big]\\ &\overset{(ii)}{\leq} \frac{\eta_k^3\tilde{L}}{nb}\sum_{i=1}^m(n-b(i-1))^2\frac{b(i-1)}{(n-b(i-1))(n-1)}\sigma_*^2\\ &= \frac{\eta_k^3\tilde{L}(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2,\end{aligned}$$

where (i) is due to the linearity of expectation, and we use our definition $\sigma_*^2=\frac{1}{n}\sum_{j=1}^n(\boldsymbol{y}_*^j)^2\|\boldsymbol{a}_j\|_2^2=\mathbb{E}_j[\|\boldsymbol{y}_*^j\boldsymbol{a}_j\|_2^2]$ for (ii). Taking expectation w.r.t. all the randomness on both sides and using the law of total expectation, we obtain

$$\mathbb{E}[\mathcal{T}_2]\leq\mathbb{E}\Big[\frac{\eta_k^3n\hat{L}_{\pi^{(k)}}\tilde{L}_{\pi^{(k)}}}{b^2}\|\boldsymbol{y}_k-\boldsymbol{y}_{*,k}\|_{\boldsymbol{\Lambda}_k^{-1}}^2+\frac{\eta_k}{2n}\|\boldsymbol{v}_k-\boldsymbol{y}_k\|_{\boldsymbol{\Lambda}_k^{-1}}^2\Big]+\frac{\eta_k^3\tilde{L}(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

For the SO scheme, since there is only one random permutation generated at the very beginning, we can take expectation w.r.t. all the randomness on both sides of (2.25), and the randomness for the term $\frac{\eta_k^3\tilde{L}}{nb}\mathbb{E}\Big[\sum_{i=1}^m\|\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_{*,k}\|_2^2\Big]$ is only from the initial random permutation. So the above argument still applies to this case, and we complete the proof. □

Theorem 3. *Under Assumptions 3 and 4, if* $\eta_k\leq\frac{b}{n\sqrt{2\hat{L}_{\pi^{(k)}}\tilde{L}_{\pi^{(k)}}}}$ *and* $H_K=\sum_{k=1}^K\eta_k$*, then the output* $\hat{\boldsymbol{x}}_K$ *of Alg. 1 with uniformly random (RR/SO) shuffling satisfies*

$$\mathbb{E}[H_K(f(\hat{\boldsymbol{x}}_K)-f(\boldsymbol{x}_*))]\leq\frac{b}{2n}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2+\sum_{k=1}^K\frac{\eta_k^3\tilde{L}(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

As a result, given $\epsilon>0$*, there exists a constant step size* $\eta_k=\eta$ *such that* $\mathbb{E}[f(\hat{\boldsymbol{x}}_K)-f(\boldsymbol{x}_*)]\leq\epsilon$ *after* $\mathcal{O}(\frac{n\sqrt{\hat{L}\tilde{L}}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{\epsilon}+\sqrt{\frac{(n-b)(n+b)}{n(n-1)}}\frac{\sqrt{n\tilde{L}}\sigma_*\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{\epsilon^{3/2}})$ *individual gradient queries.*

Proof. Combining the bounds in Lemma 8 and 9 and plugging them into Eq. (2.15), we obtain

$$\mathbb{E}[\mathcal{E}_k]\leq\mathbb{E}\Big[\Big(\frac{\eta_k^3n\hat{L}_{\pi^{(k)}}\tilde{L}_{\pi^{(k)}}}{b^2}-\frac{\eta_k}{2n}\Big)\|\boldsymbol{y}_k-\boldsymbol{y}_{*,k}\|_{\boldsymbol{\Lambda}_k^{-1}}^2\Big]+\frac{\eta_k^3\tilde{L}(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

For the stepsize η_k such that $\eta_k\leq\frac{b}{n\sqrt{2\hat{L}_{\pi^{(k)}}\tilde{L}_{\pi^{(k)}}}}$, we have $\frac{\eta_k^3n\hat{L}_{\pi^{(k)}}\tilde{L}_{\pi^{(k)}}}{b^2}-\frac{\eta_k}{2n}\leq0$, thus

$$\mathbb{E}[\mathcal{E}_k]\leq\frac{\eta_k^3\tilde{L}(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

Noticing that $\mathcal{E}_k = \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*) + \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_k\|_2^2 - \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1}\|_2^2$ and telescoping from $k = 1$ to K, we have

$$\mathbb{E}\Big[\sum_{k=1}^{K} \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*)\Big] \le \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_0\|_2^2 - \frac{b}{2n}\mathbb{E}[\|\boldsymbol{x}_* - \boldsymbol{x}_K\|_2^2] + \sum_{k=1}^{K} \frac{\eta_k^3 \tilde{L}(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

Noticing that $\mathcal{L}(\boldsymbol{x}, \boldsymbol{v})$ is convex w.r.t. $\boldsymbol{x}$, we have $\mathrm{Gap}^{\boldsymbol{v}}(\hat{\boldsymbol{x}}_K, \boldsymbol{y}_*) \le \sum_{k=1}^{K} \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*)/H_K$, where $\hat{\boldsymbol{x}}_K = \sum_{k=1}^{K} \eta_k \boldsymbol{x}_k / H_K$ and $H_K = \sum_{k=1}^{K} \eta_k$, which leads to

$$\mathbb{E}\Big[H_K \mathrm{Gap}^{\boldsymbol{v}}(\hat{\boldsymbol{x}}_K, \boldsymbol{y}_*)\Big] \le \frac{b}{2n}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^{K} \frac{\eta_k^3 \tilde{L}(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

Further choosing $\boldsymbol{v} = \boldsymbol{y}_{\hat{\boldsymbol{x}}_K}$, we obtain

$$\mathbb{E}[H_K(f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*))] \le \frac{b}{2n}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^{K} \frac{\eta_k^3 \tilde{L}(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2. \tag{2.26}$$

To analyze the individual gradient oracle complexity, we choose constant stepsizes $\eta \le \frac{b}{n\sqrt{2\hat{L}\tilde{L}}}$, then Eq. (2.26) will become

$$\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \le \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{\eta^2 \tilde{L}(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2.$$

Without loss of generality, we assume that $b \neq n$, otherwise the method and its analysis reduce to (full) gradient descent. We consider the following two cases:

- "Small K" case: if $\eta = \frac{b}{n\sqrt{2\hat{L}\tilde{L}}} \le \Big(\frac{3b^3(n-1)\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{n(n-b)(n+b)\tilde{L}K\sigma_*^2}\Big)^{1/3}$, we have

$$\begin{aligned}\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] &\le \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{\eta^2 \tilde{L}(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2 \\ &\le \frac{\sqrt{\hat{L}\tilde{L}}}{\sqrt{2}K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{1}{2}\Big(\frac{(n-b)(n+b)}{n^2(n-1)}\Big)^{1/3}\frac{\tilde{L}^{1/3}\sigma_*^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{3^{1/3}K^{2/3}}.\end{aligned}$$

- "Large K" case: if $\eta = \Big(\frac{3b^3(n-1)\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{n(n-b)(n+b)\tilde{L}K\sigma_*^2}\Big)^{1/3} \le \frac{b}{n\sqrt{2\hat{L}\tilde{L}}}$, we have

$$\begin{aligned}\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] &\le \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{\eta^2 \tilde{L}(n-b)(n+b)}{6b^2(n-1)}\sigma_*^2 \\ &\le \Big(\frac{(n-b)(n+b)}{n^2(n-1)}\Big)^{1/3}\frac{\tilde{L}^{1/3}\sigma_*^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{3^{1/3}K^{2/3}}.\end{aligned}$$

Combining these two cases by setting $\eta = \min\Big\{\frac{b}{n\sqrt{2\hat{L}\tilde{L}}}, \Big(\frac{3b^3(n-1)\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{n(n-b)(n+b)\tilde{L}K\sigma_*^2}\Big)^{1/3}\Big\}$, we obtain

$$\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \le \frac{\sqrt{\hat{L}\tilde{L}}}{\sqrt{2}K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \Big(\frac{(n-b)(n+b)}{n^2(n-1)}\Big)^{1/3}\frac{\tilde{L}^{1/3}\sigma_*^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{3^{1/3}K^{2/3}}.$$

Hence, to guarantee $\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \leq \epsilon$ for $\epsilon > 0$, the total number of individual gradient evaluations will be

$$nK \geq \max\Big\{\frac{n\sqrt{2\hat{L}\tilde{L}}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon}, \Big(\frac{(n-b)(n+b)}{n-1}\Big)^{1/2}\frac{2^{3/2}\tilde{L}^{1/2}\sigma_*\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{3^{1/2}\epsilon^{3/2}}\Big\},$$

as claimed. □

A few remarks are in order here. When $b = n$, we recover the standard guarantee of gradient descent, which serves as a sanity check as in this case the algorithm reduces to standard gradient descent. When $\epsilon = \Omega(\frac{(n-b)(n+b)\sigma_*^2}{n^2(n-1)\hat{L}})$, the resulting complexity is $\mathcal{O}(\frac{n\sqrt{\hat{L}\tilde{L}}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{\epsilon})$. Observe that this case can happen when either ϵ is large (compared to, say, $1/n$) or when σ_* is small (it is, in fact, possible for σ_* to be zero, which happens, for example, when the data rows are linearly independent). Unlike in bounds from previous work, we observe from our bounds the benefit of using shuffled SGD compared to full gradient descent, where the difference is by a factor that can be as large as $\sqrt{n}$, as we have discussed in the introduction (see also Section 2.5). When $\epsilon = \mathcal{O}(\frac{(n-b)(n+b)\sigma_*^2}{n^2(n-1)\hat{L}})$, the second term in our complexity bound dominates. In this case, when $b = 1$, we recover the state of the art results from [MKR20, NTDP$^+$21, CLY23], while for $b > 1$ our bound provides the $\Omega(\sqrt{\frac{n(n-1)}{(n-b)(n+b)}\cdot\frac{L}{\tilde{L}}})$-factor improvement, providing insights into benefits from the mini-batching strategy commonly used in practice.

2.4.3 Tighter Rates for Incremental Gradient Descent with Linear Predictors

We now provide the proof for convergence of IGD in the smooth convex settings. We first prove the following technical lemma, which bounds the inner product term $\mathcal{T}_2 := \frac{\eta_k}{n}\sum_{i=1}^m (\boldsymbol{v}^{(i)} - \boldsymbol{y}_k^{(i)})^\top \boldsymbol{A}^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i})$ without random permutations involved.

Lemma 10. *For any $k \in [K]$, the iterates $\{\boldsymbol{y}_k^{(i)}\}_{i=1}^m$ and $\{\boldsymbol{x}_{k-1,i}\}_{i=1}^{m+1}$ generated by Algorithm 2 with fixed data ordering satisfy*

$$\begin{aligned}\mathcal{T}_2 \leq\ & \frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_k - \boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 + \frac{\eta_k}{2n}\|\boldsymbol{v} - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}^{-1}}^2 \\ & + \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2\Big\}.\end{aligned} \tag{2.27}$$

Proof. Proceeding as in Lemma 9, we have

$$\begin{aligned}\mathcal{T}_2 :=& \frac{\eta_k}{n}\sum_{i=1}^{m}(\boldsymbol{v}^{(i)}-\boldsymbol{y}_k^{(i)})^\top \boldsymbol{A}^{(i)}(\boldsymbol{x}_k-\boldsymbol{x}_{k-1,i})\\ =& -\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_k, \boldsymbol{A}^\top \boldsymbol{I}_{(i)}(\boldsymbol{v}-\boldsymbol{y}_k)\Big\rangle\\ =& -\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}(\boldsymbol{y}_k-\boldsymbol{y}_*), \boldsymbol{A}^\top \boldsymbol{I}_{(i)}(\boldsymbol{v}-\boldsymbol{y}_k)\Big\rangle \qquad (2.28)\\ &-\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_*, \boldsymbol{A}^\top \boldsymbol{I}_{(i)}(\boldsymbol{v}-\boldsymbol{y}_k)\Big\rangle, \qquad (2.29)\end{aligned}$$

For both terms in Eq. (2.28) and Eq. (2.29), we use Young's inequality for $\alpha = 2\eta_k\tilde{L}_0 > 0$ and proceed as in Eq. (2.22) to obtain

$$\begin{aligned}&-\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}(\boldsymbol{y}_k-\boldsymbol{y}_*), \boldsymbol{A}^\top \boldsymbol{I}_{(i)}(\boldsymbol{v}-\boldsymbol{y}_k)\Big\rangle\\ \le& \frac{\eta_k^2\alpha}{2bn}\sum_{i=1}^{m}\|\boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}(\boldsymbol{y}_k-\boldsymbol{y}_*)\|_2^2+\frac{\eta_k^2}{2bn\alpha}\sum_{i=1}^{m}\|\boldsymbol{A}^\top \boldsymbol{I}_{(i)}(\boldsymbol{v}-\boldsymbol{y}_k)\|_2^2\\ \le& \frac{\eta_k^2 n\alpha}{2b^2}\hat{L}_0\|\boldsymbol{y}_k-\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2+\frac{\eta_k^2}{2n\alpha}\tilde{L}_0\|\boldsymbol{v}-\boldsymbol{y}_k\|_{\boldsymbol{\Lambda}^{-1}}^2\\ =& \frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_k-\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2+\frac{\eta_k}{4n}\|\boldsymbol{v}-\boldsymbol{y}_k\|_{\boldsymbol{\Lambda}^{-1}}^2 \qquad (2.30)\end{aligned}$$

and

$$\begin{aligned}&-\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\Big\langle \boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_*, \boldsymbol{A}^\top \boldsymbol{I}_{(i)}(\boldsymbol{v}-\boldsymbol{y}_k)\Big\rangle\\ \le& \frac{\eta_k^2\alpha}{2bn}\sum_{i=1}^{m}\|\boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_*\|_2^2+\frac{\eta_k^2}{2bn\alpha}\sum_{i=1}^{m}\|\boldsymbol{A}^\top \boldsymbol{I}_{(i)}(\boldsymbol{v}-\boldsymbol{y}_k)\|_2^2\\ \le& \frac{\eta_k^2\alpha}{2bn}\sum_{i=1}^{m}\|\boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_*\|_2^2+\frac{\eta_k^2}{2n\alpha}\tilde{L}_0\|\boldsymbol{v}-\boldsymbol{y}_k\|_{\boldsymbol{\Lambda}^{-1}}^2\\ =& \frac{\eta_k^3\tilde{L}_0}{nb}\sum_{i=1}^{m}\|\boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_*\|_2^2+\frac{\eta_k}{4n}\|\boldsymbol{v}-\boldsymbol{y}_k\|_{\boldsymbol{\Lambda}^{-1}}^2, \qquad (2.31)\end{aligned}$$

where again we used $\alpha = 2\eta_k\tilde{L}_0$. We then prove the term $\frac{\eta_k^3\tilde{L}_0}{nb}\sum_{i=1}^{m}\|\boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_*\|_2^2$ in Eq. (2.31) is no larger than the minimum of $\frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2$ and $\frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2$. Note that when $b = n$, we have $\boldsymbol{A}^\top \boldsymbol{I}_{(0)\uparrow}\boldsymbol{y}_* = 0$, so this term disappears. When $b < n$, the former one can be derived as in Eq.(2.22), which gives

$$\begin{aligned}\sum_{i=1}^{m}\|\boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_*\|_2^2 \le& \Big\|\boldsymbol{\Lambda}^{1/2}\Big(\sum_{i=1}^{m}\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{A}\boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}\Big)\boldsymbol{\Lambda}^{1/2}\Big\|_2\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 = mn\hat{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2\\ =& \frac{n^2}{b}\hat{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2.\end{aligned}$$

For the latter one, we notice that

$$\begin{aligned}\sum_{i=1}^{m}\|\boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_*\|_2^2 &= \sum_{i=1}^{m}\Big\|\sum_{j=b(i-1)+1}^{n}\boldsymbol{y}_*^j\boldsymbol{a}_j\Big\|_2^2\\ &= \sum_{i=0}^{m-1}\Big\|\sum_{j=bi+1}^{n}\boldsymbol{y}_*^j\boldsymbol{a}_j\Big\|_2^2\\ &= \sum_{i=1}^{m-1}\Big\|\sum_{j=bi+1}^{n}\boldsymbol{y}_*^j\boldsymbol{a}_j\Big\|_2^2 = \sum_{i=1}^{m-1}\Big\|\sum_{j=1}^{bi}\boldsymbol{y}_*^j\boldsymbol{a}_j\Big\|_2^2,\end{aligned}$$

by using the fact that $\sum_{j=1}^{n}\boldsymbol{y}_*^j\boldsymbol{a}_j = 0$. Using Young's inequality, we have

$$\begin{aligned}\sum_{i=1}^{m-1}\Big\|\sum_{j=1}^{bi}\boldsymbol{y}_*^j\boldsymbol{a}_j\Big\|_2^2 &\le \sum_{i=1}^{m-1} bi\sum_{j=1}^{bi}\|\boldsymbol{y}_*^j\boldsymbol{a}_j\|_2^2\\ &\le b(m-1)\sum_{i=1}^{m-1}\sum_{j=1}^{bi}\|\boldsymbol{y}_*^j\boldsymbol{a}_j\|_2^2\\ &= b(m-1)\sum_{i=1}^{m-1}\sum_{j=b(i-1)+1}^{bi}(m-i)\|\boldsymbol{y}_*^j\boldsymbol{a}_j\|_2^2\\ &\le b(m-1)^2\sum_{i=1}^{(m-1)b}\|\boldsymbol{y}_*^j\boldsymbol{a}_j\|_2^2.\end{aligned}$$

By the definition that $\sigma_*^2 = \frac{1}{n}\sum_{j=1}^{n}\|\boldsymbol{y}_*^j\boldsymbol{a}_j\|_2^2$ and $\sum_{i=i}^{(m-1)b}\|\boldsymbol{y}_*^j\boldsymbol{a}_j\|_2^2 \le \sum_{j=1}^{n}\|\boldsymbol{y}_*^j\boldsymbol{a}_j\|_2^2 = n\sigma_*^2$, we obtain

$$\frac{\eta_k^3\tilde{L}_0}{nb}\sum_{i=1}^{m}\|\boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_*\|_2^2 \le \frac{\eta_k^3\tilde{L}_0}{b}b(m-1)^2\sigma_*^2 = \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2. \tag{2.32}$$

Note that the bound in Eq. (2.32) equals to zero when $b = n$, which recovers the case of full gradient descent, so we have

$$\frac{\eta_k^3\tilde{L}_0}{nb}\sum_{i=1}^{m}\|\boldsymbol{A}^\top \boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_*\|_2^2 \le \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2\Big\}. \tag{2.33}$$

Combining Eq. (2.30)–(2.33), we obtain

$$\mathcal{T}_2 \le \frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_k - \boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 + \frac{\eta_k}{2n}\|\boldsymbol{v} - \boldsymbol{y}_k\|_{\boldsymbol{\Lambda}^{-1}}^2 + \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2\Big\},$$

thus finishing the proof. □

Theorem 4. *Under Assumptions 3 and 4, if* $\eta_k \le \frac{b}{n\sqrt{2\hat{L}_0\tilde{L}_0}}$ *and* $H_K = \sum_{k=1}^{K}\eta_k$*, the output* $\hat{\boldsymbol{x}}_K$ *of Alg. 2 with a fixed permutation satisfies*

$$H_K(f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)) \le \frac{b}{2n}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^{K}\min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2\Big\}.$$

As a consequence, given $\epsilon > 0$, there exists a constant step size $\eta_k = \eta$ such that $f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) \leq \epsilon$ after the number of gradient queries bounded by $\mathcal{O}\Big(\frac{n\sqrt{\hat{L}_0\tilde{L}_0}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{\epsilon} + \frac{\min\big\{\sqrt{n\hat{L}_0\tilde{L}_0}\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}, (n-b)\sqrt{\tilde{L}_0}\sigma_*\big\}\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{\epsilon^{3/2}}\Big)$.

Proof. Proceeding as in Lemmas 7 and 8, but without random permutations, we have

$$\begin{aligned}
\mathcal{E}_k &\leq \frac{\eta_k}{n}\sum_{i=1}^m \boldsymbol{y}_k^{(i)\top}\boldsymbol{A}^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i+1}) + \frac{\eta_k}{n}\sum_{i=1}^m (\boldsymbol{v}^{(i)} - \boldsymbol{y}_k^{(i)})^\top \boldsymbol{A}^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}) \\
&\quad - \frac{\eta_k}{2n}\|\boldsymbol{y}_k - \boldsymbol{v}\|_{\boldsymbol{\Lambda}^{-1}}^2 - \frac{\eta_k}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 - \frac{b}{2n}\sum_{i=1}^m \|\boldsymbol{x}_{k-1,i} - \boldsymbol{x}_{k-1,i+1}\|_2^2 \\
&\leq \frac{\eta_k}{n}\sum_{i=1}^m (\boldsymbol{v}^{(i)} - \boldsymbol{y}_k^{(i)})^\top \boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}) - \frac{\eta_k}{2n}\|\boldsymbol{y}_k - \boldsymbol{v}\|_{\boldsymbol{\Lambda}^{-1}}^2 - \frac{\eta_k}{2n}\|\boldsymbol{y}_k - \boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2. \qquad (2.34)
\end{aligned}$$

Using the bound in Lemma 10 and applying Eq. (2.27) into Eq. (2.34), we obtain

$$\mathcal{E}_k \leq \Big(\frac{\eta_k^3 n\hat{L}_0\tilde{L}_0}{b^2} - \frac{\eta_k}{2n}\Big)\|\boldsymbol{y}_k - \boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 + \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2\Big\}.$$

If $\eta_k \leq \frac{b}{n\sqrt{2\hat{L}_0\tilde{L}_0}}$, we have $\frac{\eta_k^3 n\hat{L}_0\tilde{L}_0}{b^2} - \frac{\eta_k}{2n} \leq 0$, thus

$$\mathcal{E}_k \leq \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2\Big\}.$$

Noticing that $\mathcal{E}_k = \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*) + \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_k\|_2^2 - \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1}\|_2^2$ and telescoping from $k = 1$ to K, we have

$$\begin{aligned}
&\sum_{k=1}^K \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*) \\
&\leq \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_0\|_2^2 - \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_K\|_2^2 + \sum_{k=1}^K \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2\Big\}.
\end{aligned}$$

Noticing that $\boldsymbol{\mathcal{L}}(\boldsymbol{x}, \boldsymbol{v})$ is convex w.r.t. $\boldsymbol{x}$, we have $\mathrm{Gap}^{\boldsymbol{v}}(\hat{\boldsymbol{x}}_K, \boldsymbol{y}_*) \leq \sum_{k=1}^K \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*)/H_K$, where $\hat{\boldsymbol{x}}_K = \sum_{k=1}^K \eta_k \boldsymbol{x}_k / H_K$ and $H_K = \sum_{k=1}^K \eta_k$, so we obtain

$$H_K \mathrm{Gap}^{\boldsymbol{v}}(\hat{\boldsymbol{x}}_K, \boldsymbol{y}_*) \leq \frac{b}{2n}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^K \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2\Big\},$$

Further choosing $\boldsymbol{v} = \boldsymbol{y}_{\hat{\boldsymbol{x}}_K}$, we obtain

$$H_K(f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)) \leq \frac{b}{2n}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^K \min\Big\{\frac{\eta_k^3 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta_k^3(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2\Big\}. \qquad (2.35)$$

To analyze the individual gradient oracle complexity, we choose constant stepsizes $\eta \leq \frac{b}{n\sqrt{2\hat{L}_0\tilde{L}_0}}$ and assume $b < n$ without loss of generality, then Eq. (2.35) becomes

$$f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) \leq \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \min\Big\{\frac{\eta^2 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2, \frac{\eta^2(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2\Big\}.$$

When $\hat{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 \leq \frac{(n-b)^2}{n}\sigma_*^2$, we set $\eta = \min\Big\{\frac{b}{n\sqrt{2\hat{L}_0\tilde{L}_0}}, \Big(\frac{b^3\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{2n^2\hat{L}_0\tilde{L}_0K\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2}\Big)^{1/3}\Big\}$ and consider the following two possible cases:

- "Small K" case: if $\eta = \frac{b}{n\sqrt{2\hat{L}_0\tilde{L}_0}} \leq \Big(\frac{b^3\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{2n^2\hat{L}_0\tilde{L}_0K\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2}\Big)^{1/3}$, we have

$$\begin{aligned} f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) &\leq \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{\eta^2 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 \\ &\leq \frac{\sqrt{\hat{L}_0\tilde{L}_0}}{\sqrt{2}K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{\hat{L}_0^{1/3}\tilde{L}_0^{1/3}\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{2^{2/3}n^{1/3}K^{2/3}}. \end{aligned}$$

- "Large K" case: if $\eta = \Big(\frac{b^3\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{2n^2\hat{L}_0\tilde{L}_0K\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2}\Big)^{1/3} \leq \frac{b}{\sqrt{2\hat{L}_0\tilde{L}_0}}$, we have

$$\begin{aligned} f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) &\leq \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{\eta^2 n}{b^2}\hat{L}_0\tilde{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2 \\ &\leq \frac{2^{1/3}\hat{L}_0^{1/3}\tilde{L}_0^{1/3}\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{n^{1/3}K^{2/3}}. \end{aligned}$$

Combining these two cases, we have

$$f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) \leq \frac{\sqrt{\hat{L}_0\tilde{L}_0}}{\sqrt{2}K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{2^{1/3}\hat{L}_0^{1/3}\tilde{L}_0^{1/3}\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{n^{1/3}K^{2/3}}.$$

Hence, to guarantee $\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \leq \epsilon$ for $\epsilon > 0$, the total number of individual gradient evaluations will be

$$nK \geq \max\Big\{\frac{n\sqrt{2\hat{L}_0\tilde{L}_0}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon}, \frac{4n^{1/2}\hat{L}_0^{1/2}\tilde{L}_0^{1/2}\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon^{3/2}}\Big\}. \tag{2.36}$$

When $\frac{(n-b)^2}{n}\sigma_*^2 \leq \hat{L}_0\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}^2$, we set $\eta = \min\Big\{\frac{b}{n\sqrt{2\hat{L}_0\tilde{L}_0}}, \Big(\frac{b^3\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{2n(n-b)^2\tilde{L}_0K\sigma_*^2}\Big)^{1/3}\Big\}$ and consider the two cases as below:

- "Small K" case: if $\eta = \frac{b}{n\sqrt{2\hat{L}_0\tilde{L}_0}} \leq \Big(\frac{b^3\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{2n(n-b)^2\tilde{L}_0K\sigma_*^2}\Big)^{1/3}$, we have

$$\begin{aligned} f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) &\leq \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{\eta^2(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2 \\ &\leq \frac{\sqrt{\hat{L}_0\tilde{L}_0}}{\sqrt{2}K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{(n-b)^{2/3}\tilde{L}_0^{1/3}\sigma_*^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{2^{2/3}n^{2/3}K^{2/3}}. \end{aligned}$$

- "Large K" case: if $\eta = \left(\frac{b^3\|\boldsymbol{x}_0-\boldsymbol{x}_*\|_2^2}{2n(n-b)^2\tilde{L}_0K\sigma_*^2}\right)^{1/3} \leq \frac{b}{n\sqrt{2\hat{L}_0\tilde{L}_0}}$, we have

$$\begin{aligned} f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) &\leq \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{\eta^2(n-b)^2}{b^2}\tilde{L}_0\sigma_*^2 \\ &\leq \frac{2^{1/3}(n-b)^{2/3}\tilde{L}_0^{1/3}\sigma_*^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{n^{2/3}K^{2/3}}. \end{aligned}$$

Combining these two cases, we obtain

$$f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*) \leq \frac{\sqrt{\hat{L}_0\tilde{L}_0}}{\sqrt{2}K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{2^{1/3}(n-b)^{2/3}\tilde{L}_0^{1/3}\sigma_*^{2/3}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^{4/3}}{n^{2/3}K^{2/3}}.$$

To guarantee $\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \leq \epsilon$ for $\epsilon > 0$, the total number of individual gradient evaluations will be

$$nK \geq \max\left\{\frac{n\sqrt{2\hat{L}_0\tilde{L}_0}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon}, \frac{4(n-b)\tilde{L}_0^{1/2}\sigma_*\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon^{3/2}}\right\}. \tag{2.37}$$

Combining Eq. (2.36) and Eq. (2.37), we finally have

$$\begin{aligned} nK \geq\ & \frac{n\sqrt{2\hat{L}_0\tilde{L}_0}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon} \\ & + \min\left\{\frac{4n^{1/2}\hat{L}_0^{1/2}\tilde{L}_0^{1/2}\|\boldsymbol{y}_*\|_{\boldsymbol{\Lambda}^{-1}}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon^{3/2}}, \frac{4(n-b)\tilde{L}_0^{1/2}\sigma_*\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon^{3/2}}\right\}, \end{aligned}$$

thus finishing the proof. □

2.4.4 Extension to non-smooth convex loss functions

In non-smooth settings, we make the following standard assumption.

Assumption 5. *Each ℓ_i is convex and G_i-Lipschitz ($i \in [n]$), i.e., $|\ell_i(x) - \ell_i(y)| \leq G_i|x - y|$ for any $x, y \in \mathbb{R}$; thus $|g_i(x)| \leq G_i$ where $g_i(x) \in \partial\ell_i(x)$. There exists a minimizer $\boldsymbol{x}_* \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$.*

If Assumption 5 holds, each $\ell_i(\boldsymbol{a}_i^\top\boldsymbol{x})$ is also $G_{\max}$-Lipschitz with respect to $\boldsymbol{x}$, where $G_{\max} = \max_{i\in[n]} G_i\|\boldsymbol{a}_i\|_2$. To state our results, we define $\boldsymbol{\Gamma} := \mathrm{diag}(G_1^2, G_2^2, \ldots, G_n^2)$ and $\boldsymbol{\Gamma}_\pi = \mathrm{diag}(G_{\pi_1}^2, G_{\pi_2}^2, \ldots, G_{\pi_n}^2)$, given a data permutation π of $[n]$.

We now extend our analysis of Algorithm 1 to convex nonsmooth Lipschitz settings, where the conjugate functions $\ell_i^*(y^i)$ are only convex. Proceeding as in Lemma 4, we obtain a bound on the primal-dual gap similar to (2.2), but lose two retraction terms induced by smoothness. Instead of cancelling the corresponding error terms like in the smooth case, we rely on the boundedness of the subgradients to bound these terms under a sufficiently small step size, which is common in nonsmooth

Lipschitz settings. Similar to Section 2.3, we introduce the following quantities to obtain a tighter guarantee with respect to the data matrix and Lipschitz constants

$$\hat{G}_\pi := \frac{1}{mn}\|\boldsymbol{\Gamma}_\pi^{1/2}(\textstyle\sum_{j=1}^m \boldsymbol{I}_{b(j-1)\uparrow}\boldsymbol{A}_\pi\boldsymbol{A}_\pi^\top\boldsymbol{I}_{b(j-1)\uparrow})\boldsymbol{\Gamma}_\pi^{1/2}\|_2,$$
$$\tilde{G}_\pi := \frac{1}{b}\|\boldsymbol{\Gamma}_\pi^{1/2}(\textstyle\sum_{j=1}^m \boldsymbol{I}_{(j)}\boldsymbol{A}_\pi\boldsymbol{A}_\pi^\top\boldsymbol{I}_{(j)})\boldsymbol{\Gamma}_\pi^{1/2}\|_2.$$

We discuss the improvements in convergence from $\hat{G}_\pi$ and $\tilde{G}_\pi$ in Section 2.5, while we prove the convergence of Algorithm 2 in the non-smooth convex settings in the following. we first recall the following standard first-order characterization of convexity.

Lemma 11. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuous convex function. Then, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$:*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{g}_{\boldsymbol{x}}, \boldsymbol{y} - \boldsymbol{x}\rangle,$$

where $\boldsymbol{g}_{\boldsymbol{x}} \in \partial f(\boldsymbol{x})$, and $\partial f(\boldsymbol{x})$ is the subdifferential of f at $\boldsymbol{x}$.

The following technical lemma provides a primal-dual gap bound in convex nonsmooth settings.

Lemma 12. *For any $k \in [K]$, the iterates $\{\boldsymbol{y}_k^{(i)}\}_{i=1}^m$ and $\{\boldsymbol{x}_{k-1,i}\}_{i=1}^{m+1}$ generated by Algorithm 2 satisfy*

$$\begin{aligned}\mathcal{E}_k \leq \frac{\eta_k}{n}\sum_{i=1}^m \Big(\boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i+1}) + (\boldsymbol{v}_k^{(i)} - \boldsymbol{y}_k^{(i)})^\top\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i})\Big) \\ - \frac{b}{2n}\sum_{i=1}^m \|\boldsymbol{x}_{k-1,i} - \boldsymbol{x}_{k-1,i+1}\|_2^2,\end{aligned} \tag{2.38}$$

where $\mathcal{E}_k := \eta_k(\mathcal{L}(\boldsymbol{x}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*)) + \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_k\|_2^2 - \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1}\|_2^2$.

Proof. By the same argument as in the proof for Lemma 7, we know that $\boldsymbol{a}_{\pi_j^{(k)}}^\top \boldsymbol{x}_{k-1,i} \in \partial \ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j)$ for $b(i-1)+1 \leq j \leq bi$, then by Lemma 11 we have

$$\ell^*_{\pi_j^{(k)}}(\boldsymbol{v}_k^j) \geq \ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j) + \boldsymbol{a}_{\pi_j^{(k)}}^\top \boldsymbol{x}_{k-1,i}(\boldsymbol{v}_k^j - \boldsymbol{y}_k^j),$$

which leads to

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}_k, \boldsymbol{v}) &= \frac{1}{n}\sum_{i=1}^m \Big(\boldsymbol{v}_k^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_{k-1,i} - \sum_{j=b(i-1)+1}^{bi} \ell^*_{\pi_j^{(k)}}(\boldsymbol{v}_k^j)\Big) + \frac{1}{n}\sum_{i=1}^m \boldsymbol{v}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}) \\ &\leq \frac{1}{n}\sum_{i=1}^m \Big(\boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_{k-1,i} - \sum_{j=b(i-1)+1}^{bi} \ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_k^j)\Big) + \frac{1}{n}\sum_{i=1}^m \boldsymbol{v}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1,i}).\end{aligned} \tag{2.39}$$

Using the same argument for $\mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*)$ as $\boldsymbol{a}_j^\top \boldsymbol{x}_* \in \partial \ell_j^*(\boldsymbol{y}_*^j)$ for $j \in [n]$, we have

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*) &= \frac{1}{n}\sum_{i=1}^{m}\Big(\boldsymbol{y}_{*,k}^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_* - \sum_{j=b(i-1)+1}^{bi}\ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_{*,k}^j)\Big)\\ &\geq \frac{1}{n}\sum_{i=1}^{m}\Big(\boldsymbol{y}_{k}^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_* - \sum_{j=b(i-1)+1}^{bi}\ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_{k}^j)\Big). \end{aligned} \tag{2.40}$$

Adding and substracting the term $\frac{b}{2n\eta_k}\sum_{i=1}^m\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|_2^2$ on the R.H.S. of Eq. (2.40), we obtain

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*) \geq \frac{1}{n}\sum_{i=1}^{m}\Big(\boldsymbol{y}_{k}^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_* + \frac{b}{2\eta_k}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|_2^2 - \sum_{j=b(i-1)+1}^{bi}\ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_{k}^j)\Big)\\ - \frac{b}{2n\eta_k}\sum_{i=1}^{m}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|_2^2.\end{aligned}$$

Denote $\phi_k^{(i)}(\boldsymbol{x}) := \boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x} + \frac{b}{2\eta_k}\|\boldsymbol{x} - \boldsymbol{x}_{k-1,i}\|_2^2$, which is $\frac{b}{\eta_k}$-strongly convex w.r.t. $\boldsymbol{x}$. Noticing that $\boldsymbol{x}_{k-1,i+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d}\left\{\boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x} + \frac{b}{2\eta_k}\|\boldsymbol{x} - \boldsymbol{x}_{k-1,i}\|^2\right\}$ by Line 7 of Alg. 2, we have $\nabla\phi_k^{(i)}(\boldsymbol{x}_{k-1,i+1}) = \boldsymbol{0}$, which leads to

$$\phi_k^{(i)}(\boldsymbol{x}_*) \geq \phi_k^{(i)}(\boldsymbol{x}_{k-1,i+1}) + \frac{b}{2\eta_k}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i+1}\|_2^2.$$

Thus, we obtain

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*) &\geq \frac{1}{n}\sum_{i=1}^{m}\Big(\boldsymbol{y}_{k}^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_{k-1,i+1} + \frac{b}{2\eta_k}\|\boldsymbol{x}_{k-1,i+1} - \boldsymbol{x}_{k-1,i}\|_2^2 - \sum_{j=b(i-1)+1}^{bi}\ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_{k}^j)\Big)\\ &\quad + \frac{b}{2n\eta_k}\sum_{i=1}^{m}\left(\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i+1}\|_2^2 - \|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|_2^2\right)\\ &\overset{(i)}{=} \frac{1}{n}\sum_{i=1}^{m}\Big(\boldsymbol{y}_{k}^{(i)\top}\boldsymbol{A}_k^{(i)}\boldsymbol{x}_{k-1,i+1} + \frac{b}{2\eta_k}\|\boldsymbol{x}_{k-1,i+1} - \boldsymbol{x}_{k-1,i}\|_2^2 - \sum_{j=b(i-1)+1}^{bi}\ell^*_{\pi_j^{(k)}}(\boldsymbol{y}_{k}^j)\Big)\\ &\quad + \frac{b}{2n\eta_k}\|\boldsymbol{x}_k - \boldsymbol{x}_*\|_2^2 - \frac{b}{2n\eta_k}\|\boldsymbol{x}_{k-1} - \boldsymbol{x}_*\|_2^2,\end{aligned} \tag{2.41}$$

where (i) is by telescoping $\sum_{i=1}^m\left(\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i+1}\|_2^2 - \|\boldsymbol{x}_* - \boldsymbol{x}_{k-1,i}\|_2^2\right)$ and using $\boldsymbol{x}_k = \boldsymbol{x}_{k-1,m+1}$ and $\boldsymbol{x}_{k-1} = \boldsymbol{x}_{k-1,1}$, which both hold by definition.

Combining the bounds from Eq. (2.39) and Eq. (2.41), and denoting

$$\mathcal{E}_k := \eta_k(\mathcal{L}(\boldsymbol{x}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{x}_*, \boldsymbol{y}_*)) + \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_k\|_2^2 - \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1}\|_2^2,$$

we finally obtain

$$\begin{aligned}\mathcal{E}_k &\leq \frac{\eta_k}{n}\sum_{i=1}^{m}\boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_{k-1,i}-\boldsymbol{x}_{k-1,i+1})+\frac{\eta_k}{n}\sum_{i=1}^{m}\boldsymbol{v}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k-\boldsymbol{x}_{k-1,i})\\&\quad-\frac{b}{2n}\sum_{i=1}^{m}\|\boldsymbol{x}_{k-1,i}-\boldsymbol{x}_{k-1,i+1}\|_2^2\\&=\frac{\eta_k}{n}\sum_{i=1}^{m}\boldsymbol{y}_k^{(i)\top}\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k-\boldsymbol{x}_{k-1,i+1})+\frac{\eta_k}{n}\sum_{i=1}^{m}(\boldsymbol{v}_k^{(i)}-\boldsymbol{y}_k^{(i)})^\top\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k-\boldsymbol{x}_{k-1,i})\\&\quad-\frac{b}{2n}\sum_{i=1}^{m}\|\boldsymbol{x}_{k-1,i}-\boldsymbol{x}_{k-1,i+1}\|_2^2,\end{aligned}$$

thus completing the proof. □

Note that we can still use Lemma 8 to bound the first inner product term in Eq. (2.38), as we are studying the same algorithm. The following lemma provides a bound on the second inner product term $\mathcal{T}_2 := \frac{\eta_k}{n}\sum_{i=1}^{m}(\boldsymbol{v}_k^{(i)}-\boldsymbol{y}_k^{(i)})^\top\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k-\boldsymbol{x}_{k-1,i})$ in Eq. (2.38).

Lemma 13. *Under Assumption 5, for any $k\in[K]$, the iterates $\{\boldsymbol{y}_k^{(i)}\}_{i=1}^m$ and $\{\boldsymbol{x}_{k-1,i}\}_{i=1}^{m+1}$ generated by Algorithm 2 satisfy*

$$\mathcal{T}_2 \leq \frac{\eta_k^2\sqrt{\hat{G}_{\pi^{(k)}}\tilde{G}_{\pi^{(k)}}}}{b}\|\boldsymbol{y}_k\|_{\boldsymbol{\Gamma}_k^{-1}}^2+\frac{\eta_k^2\sqrt{\hat{G}_{\pi^{(k)}}\tilde{G}_{\pi^{(k)}}}}{4b}\|\boldsymbol{v}_k-\boldsymbol{y}_k\|_{\boldsymbol{\Gamma}_k^{-1}}^2. \tag{2.42}$$

Proof. Proceeding as in Lemma 9, we have

$$\mathcal{T}_2 := \frac{\eta_k}{n}\sum_{i=1}^{m}(\boldsymbol{v}_k^{(i)}-\boldsymbol{y}_k^{(i)})^\top\boldsymbol{A}_k^{(i)}(\boldsymbol{x}_k-\boldsymbol{x}_{k-1,i}) = -\frac{\eta_k^2}{bn}\sum_{i=1}^{m}\left\langle\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_k,\boldsymbol{A}_k^\top\boldsymbol{I}_{(i)}(\boldsymbol{v}_k-\boldsymbol{y}_k)\right\rangle.$$

Using Young's inequality for some $\alpha>0$ and proceeding as in Eq. (2.22), we obtain

$$\begin{aligned}\mathcal{T}_2 &\leq \frac{\eta_k^2\alpha}{2bn}\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top\boldsymbol{I}_{b(i-1)\uparrow}\boldsymbol{y}_k\|_2^2+\frac{\eta_k^2}{2bn\alpha}\sum_{i=1}^{m}\|\boldsymbol{A}_k^\top\boldsymbol{I}_{(i)}(\boldsymbol{v}_k-\boldsymbol{y}_k)\|_2^2\\&\leq\frac{\eta_k^2 n\alpha}{2b^2}\hat{G}_{\pi^{(k)}}\|\boldsymbol{y}_k\|_{\boldsymbol{\Gamma}_k^{-1}}^2+\frac{\eta_k^2}{2n\alpha}\tilde{G}_{\pi^{(k)}}\|\boldsymbol{v}_k-\boldsymbol{y}_k\|_{\boldsymbol{\Gamma}_k^{-1}}^2,\end{aligned}$$

where we use our definitions that $\hat{G}_{\pi^{(k)}} := \frac{1}{mn}\|\boldsymbol{\Gamma}_k^{1/2}(\sum_{j=1}^{m}\boldsymbol{I}_{b(j-1)\uparrow}\boldsymbol{A}_k\boldsymbol{A}_k^\top\boldsymbol{I}_{b(j-1)\uparrow})\boldsymbol{\Gamma}_k^{1/2}\|_2$ and $\tilde{G}_{\pi^{(k)}} := \frac{1}{b}\|\boldsymbol{\Gamma}_k^{1/2}(\sum_{j=1}^{m}\boldsymbol{I}_{(j)}\boldsymbol{A}_k\boldsymbol{A}_k^\top\boldsymbol{I}_{(j)})\boldsymbol{\Gamma}_k^{1/2}\|_2$. It remains to choose $\alpha=\frac{2b}{n}\sqrt{\frac{\tilde{G}_k}{\hat{G}_k}}$ to finish the proof. □

We are now ready to prove Theorem 5 for the convergence of shuffled SGD in the convex nonsmooth Lipschitz settings.

Theorem 5. *Under Assumption 5, if* $H_K = \sum_{k=1}^K \eta_k$ *and* $\bar{G} = \mathbb{E}_\pi[\sqrt{\hat{G}_\pi \tilde{G}_\pi}]$*, the output* $\hat{\boldsymbol{x}}_K$ *of Alg. 1 with possible uniformly random shuffling satisfies*

$$\mathbb{E}[H_K(f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*))] \leq \frac{1}{2n}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^K 2\eta_k^2 n\bar{G},$$

As a result, for any $\epsilon > 0$*, there exists a step size* $\eta_k = \eta$ *such that* $\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \leq \epsilon$ *after* $\mathcal{O}(\frac{n\bar{G}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon^2})$ *individual gradient queries.*

Proof. To simplify the presentation of our analysis, we first assume $\|\boldsymbol{v}\|_{\boldsymbol{\Gamma}^{-1}}^2 \leq n$, which will be later verified by our choice of $\boldsymbol{v} = \boldsymbol{y}_{\hat{\boldsymbol{x}}_K}$ and Assumption 5.

Combining the bounds in Lemma 8 and 13 and plugging them into Eq. (2.38), we have

$$\begin{aligned}
\mathcal{E}_k &\leq \frac{\eta_k^2\sqrt{\hat{G}_{\pi^{(k)}}\tilde{G}_{\pi^{(k)}}}}{b}\|\boldsymbol{y}_k\|_{\boldsymbol{\Gamma}_k^{-1}}^2 + \frac{\eta_k^2\sqrt{\hat{G}_{\pi^{(k)}}\tilde{G}_{\pi^{(k)}}}}{4b}\|\boldsymbol{v}_k - \boldsymbol{y}_k\|_{\boldsymbol{\Gamma}_k^{-1}}^2 \\
&\overset{(i)}{\leq} \frac{\eta_k^2\sqrt{\hat{G}_{\pi^{(k)}}\tilde{G}_{\pi^{(k)}}}}{b}\|\boldsymbol{y}_k\|_{\boldsymbol{\Gamma}_k^{-1}}^2 + \frac{\eta_k^2\sqrt{\hat{G}_{\pi^{(k)}}\tilde{G}_{\pi^{(k)}}}}{2b}(\|\boldsymbol{v}\|_{\boldsymbol{\Gamma}^{-1}}^2 + \|\boldsymbol{y}_k\|_{\boldsymbol{\Gamma}_k^{-1}}^2) \\
&\overset{(ii)}{\leq} \frac{2\eta_k^2 n\sqrt{\hat{G}_{\pi^{(k)}}\tilde{G}_{\pi^{(k)}}}}{b},
\end{aligned} \tag{2.43}$$

where we use Young's inequality for $\|\boldsymbol{v}_k - \boldsymbol{y}_k\|_{\boldsymbol{\Gamma}_k^{-1}}^2$ and $\|\boldsymbol{v}_k\|_{\boldsymbol{\Gamma}_k^{-1}} = \|\boldsymbol{v}\|_{\boldsymbol{\Gamma}^{-1}}^2$ as $\boldsymbol{v}$ is a fixed vector for (i), and (ii) is due to $\|\boldsymbol{y}_k\|_{\boldsymbol{\Gamma}_k^{-1}}^2 \leq n$ by Assumption 5 and assuming that $\|\boldsymbol{v}\|_{\boldsymbol{\Gamma}^{-1}}^2 \leq n$. Proceeding as the proof for Theorem 3, we first assume the RR scheme and take conditional expectation w.r.t. the randomness up to but not including k-th epoch, then we obtain

$$\mathbb{E}_k[\mathcal{E}_k] \leq \frac{2\eta_k^2 n\mathbb{E}_k[\sqrt{\hat{G}_{\pi^{(k)}}\tilde{G}_{\pi^{(k)}}}]}{b}.$$

Since the randomness only comes from the random permutation $\pi^{(k)}$, we have

$$\mathbb{E}_k[\mathcal{E}_k] \leq \frac{2\eta_k^2 n\mathbb{E}_\pi[\sqrt{\hat{G}_\pi\tilde{G}_\pi}]}{b}.$$

For notational convenience, we denote $\bar{G} = \mathbb{E}_\pi[\sqrt{\hat{G}_\pi\tilde{G}_\pi}]$, and further take expectation w.r.t. all the randomness on both sides and use the law of total expectation to obtain

$$\mathbb{E}[\mathcal{E}_k] \leq \frac{2\eta_k^2 n\bar{G}}{b}. \tag{2.44}$$

For the SO scheme, there is one random permutation π generated at the very beginning such that $\pi^{(k)} = \pi$ for all $k \in [K]$. So we can directly take expectation w.r.t. all the randomness on both sides of Eq. (2.43), with the randomness only from π, which leads to the same bound as Eq. (2.44)

with $\mathbb{E}[\sqrt{\hat{G}_{\pi^{(k)}}\tilde{G}_{\pi^{(k)}}}] = \mathbb{E}_\pi[\sqrt{\hat{G}_\pi \tilde{G}_\pi}]$. Note that for incremental gradient (IG) descent, we can let $\bar{G} = \sqrt{\hat{G}_0\tilde{G}_0}$ without randomness involved, where $\hat{G}_0 = \hat{G}_{\pi^{(0)}}$ and $\tilde{G}_0 = \tilde{G}_{\pi^{(0)}}$ w.r.t. the initial, fixed permutation $\pi^{(0)}$ of the data matrix $\boldsymbol{A}$.

Noticing that $\mathcal{E}_k = \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*) + \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_k\|_2^2 - \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_{k-1}\|_2^2$ and telescoping from $k = 1$ to K, we have

$$\mathbb{E}\Big[\sum_{k=1}^K \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*)\Big] \leq \frac{b}{2n}\|\boldsymbol{x}_* - \boldsymbol{x}_0\|_2^2 - \frac{b}{2n}\mathbb{E}[\|\boldsymbol{x}_* - \boldsymbol{x}_K\|_2^2] + \sum_{k=1}^K \frac{2\eta_k^2 n\bar{G}}{b}.$$

Noticing that $\boldsymbol{\mathcal{L}}(\boldsymbol{x}, \boldsymbol{v})$ is convex wrt $\boldsymbol{x}$, we have $\mathrm{Gap}^{\boldsymbol{v}}(\hat{\boldsymbol{x}}_K, \boldsymbol{y}_*) \leq \sum_{k=1}^K \eta_k \mathrm{Gap}^{\boldsymbol{v}}(\boldsymbol{x}_k, \boldsymbol{y}_*)/H_K$, where $\hat{\boldsymbol{x}}_K = \sum_{k=1}^K \eta_k \boldsymbol{x}_k / H_K$ and $H_K = \sum_{k=1}^K \eta_k$, so we obtain

$$\mathbb{E}\Big[H_K \mathrm{Gap}^{\boldsymbol{v}}(\hat{\boldsymbol{x}}_K, \boldsymbol{y}_*)\Big] \leq \frac{b}{2n}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^K \frac{2\eta_k^2 n\bar{G}}{b}.$$

Further choosing $\boldsymbol{v} = \boldsymbol{y}_{\hat{\boldsymbol{x}}_K}$, which also verifies $\|\boldsymbol{v}\|_{\boldsymbol{\Gamma}^{-1}}^2 = \|\boldsymbol{y}_{\hat{\boldsymbol{x}}_K}\|_{\boldsymbol{\Gamma}^{-1}}^2 \leq n$ by Assumption 5, we obtain

$$\mathbb{E}[H_K(f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*))] \leq \frac{b}{2n}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \sum_{k=1}^K \frac{2\eta_k^2 n\bar{G}}{b}.$$

To analyze the individual gradient oracle complexity, we choose constant stepsize η. Then, the above bound becomes

$$\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \leq \frac{b}{2n\eta K}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2 + \frac{2n\eta\bar{G}}{b}.$$

Choosing $\eta = \frac{b\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2}{2n\sqrt{K\bar{G}}}$, we have

$$\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \leq \frac{2\sqrt{\bar{G}}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2}{\sqrt{K}}.$$

Hence, given $\epsilon > 0$, to ensure $\mathbb{E}[f(\hat{\boldsymbol{x}}_K) - f(\boldsymbol{x}_*)] \leq \epsilon$, the total number of individual gradient evaluations will be

$$nK \geq \frac{4n\bar{G}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon^2},$$

thus completing the proof. □

We now briefly discuss this result. The total number of individual gradient queries is $\mathcal{O}(\frac{n\bar{G}\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|_2^2}{\epsilon^2})$, which appears independent of the batch size, but this is actually not the case, as the parameter $\bar{G} = \mathbb{E}_\pi[\sqrt{\hat{G}_\pi \tilde{G}_\pi}]$ depends on the block partitioning, due to Eq. (2.38). When $b = n$, as a sanity check, we recover the standard guarantee of (full) subgradient descent, which is expected, as in this case shuffled SGD reduces to subgradient descent. When $b = 1$, however, the bound is worse than the corresponding bound for standard SGD, by a factor $\mathcal{O}(n\bar{G}/G^2)$. By a similar sequence of

inequalities as in Eq. (2.45), this factor is never worse than n, but it is typically much smaller, taking values as small as 1. We note that it is not known whether a better bound is possible for shuffled SGD in this setting, as the only seemingly tighter upper bound from [Sha16] applies only for constant K, when $n = \Omega(\frac{1}{\epsilon^2})$, and under an additional boundedness assumption for the algorithm iterates.

2.5 Discussion of Our New Smoothness Constants and Numerical Results

To succinctly explain where our improvements come from, we now consider (PL) where ℓ_i is 1-smooth and $b = 1$, ignoring the gains from the mini-batch estimators (for large K) and our softer guarantee that handles individual smoothness constants. For this specific case, $\tilde{L} = L_{\max} = \max_{1\leq i\leq n}\|\boldsymbol{a}_i\|^2$, and thus our results for the smooth case and the RR and SO variants match state of the art in the second term, which dominates when there are many ($K = \Omega(\frac{L_{\max}^2 D^2 n}{\sigma_*^2})$) epochs. When there are $K = O(\frac{L_{\max}^2 D^2 n}{\sigma_*^2})$ epochs in the SO and RR variants or for all regimes of K in the IG variant, the difference between our and state of the art bounds comes from the constant $\hat{L}$ that replaces $L_{\max}$, and our improvement is by a factor $\sqrt{L_{\max}/\hat{L}}$. Note that $\mathcal{O}(\frac{nL_{\max}}{\epsilon})$ from prior bounds, which is the dominating term in the small K regime, is even worse than the complexity of full gradient descent, as the full gradient Lipschitz constant of f in this case is $\frac{1}{n}\|\boldsymbol{A}\boldsymbol{A}^\top\|_2 \leq L_{\max}$.

Given a worst-case permutation $\bar{\pi}$, and denoting by $\boldsymbol{A}_{\bar{\pi}}$ the data matrix $\boldsymbol{A}$ with its rows permuted according to $\bar{\pi}$, our constant $\hat{L}$ can be bounded above by $L_{\max}$ using the following sequence of inequalities:

$$\begin{aligned}\hat{L} = \tfrac{1}{n^2}\|\textstyle\sum_{j=1}^n \boldsymbol{I}_{(j-1)\uparrow}\boldsymbol{A}_{\bar{\pi}}\boldsymbol{A}_{\bar{\pi}}^\top\boldsymbol{I}_{(j-1)\uparrow}\|_2 &\overset{(i)}{\leq} \tfrac{1}{n^2}\textstyle\sum_{j=1}^n\|\boldsymbol{I}_{(j-1)\uparrow}\boldsymbol{A}_{\bar{\pi}}\boldsymbol{A}_{\bar{\pi}}^\top\boldsymbol{I}_{(j-1)\uparrow}\|_2 \\ &\overset{(ii)}{\leq} \tfrac{1}{n^2}\textstyle\sum_{j=1}^n\|\boldsymbol{A}_{\bar{\pi}}\boldsymbol{A}_{\bar{\pi}}^\top\|_2 \\ &\overset{(iii)}{\leq} \tfrac{1}{n}\textstyle\sum_{i=1}^n\|\boldsymbol{a}_i\|_2^2 \leq \max_{1\leq i\leq n}\|\boldsymbol{a}_i\|_2^2 = L_{\max},\end{aligned} \tag{2.45}$$

where (i) holds by the triangle inequality, (ii) holds because the operator norm of the matrix $\boldsymbol{I}_{(j-1)\uparrow}\boldsymbol{A}_\pi\boldsymbol{A}_\pi^\top\boldsymbol{I}_{(j-1)\uparrow}$ (equal to the operator norm of the bottom right $(n-j+1)\times(n-j+1)$ submatrix of $\boldsymbol{A}_\pi\boldsymbol{A}_\pi^\top$) is always at most $\|\boldsymbol{A}_\pi\boldsymbol{A}_\pi^\top\| = \|\boldsymbol{A}\boldsymbol{A}^\top\|$, for any permutation π, and (iii) holds by bounding above the operator norm of a symmetric matrix by its trace. Hence $\hat{L}$ is never larger than $L_{\max}$, but can generally be much smaller, due to the sequence of inequality relaxations in (2.45). While each of these inequalities can be loose, we emphasize that (iii) is almost always loose, by a factor that can be as large as n.

As a specific example where $\hat{L}$ is smaller than $L_{\max}$ by a factor of n, consider the example of Gaussian data, where we draw n i.i.d. standard Gaussian vectors from $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and take $d = n$. By standard concentration results, with high probability, all columns/rows of $\mathbf{A}_{\bar{\pi}}$ in this case are

near-orthogonal (see, e.g., [BHK20, Chapter]) and $\|\boldsymbol{a}_i\|_2^2 \approx d = n$ for all i. As a result, the operator norm to trace inequality (iii) is loose by a factor $d = n$, with high probability. Note that in this example all individual smoothness parameters of components f_i are essentially the same (w.h.p.) and equal $\|\boldsymbol{a}_i\|_2^2$, thus the improvement of our bound on the smoothness parameter does not come from averaging but from the structure of the data. This observation is important for contrasting the results from Section 2.3 and Section 2.4. In particular, focusing solely on the finite sum structure and ignoring the structure of the data matrix would provide no improvements in the resulting convergence bounds.

As further evidence, we empirically evaluate $L_{\max}/\hat{L}$ on 15 large-scale machine learning datasets and demonstrate that on those datasets $L_{\max}/\hat{L}$ is of the order n^{α}, for $\alpha \in [0.15, 0.96]$ (see Sec. 2.5.1 for more details), providing strong evidence of a tighter guarantee as a function of n.

For the nonsmooth settings, by a similar sequence of inequalities, we can show that $\bar{G} \leq G_{\max}^2$, which can be loose by a factor $1/n$ due to the operator norm to trace inequality. Thus, our bound is never worse than what would be obtained from the full subgradient method, but can match the bound of standard SGD, or even improve[1] upon it for at least some data matrices $\boldsymbol{A}$.

2.5.1 Numerical results and discussion

In this section, we provide empirical evidence to support our claim about usefulness of the new convergence bounds obtained in our work. In particular, we conduct numerical evaluations to compare $\hat{L}$ to the classical smoothness constant L on synthetic datasets and on popular machine learning benchmark datasets.

For a more streamlined comparison and to focus on the dependence on the data matrix, we assume that the loss functions ℓ_i all have the same smoothness constant, which leads to $L_{\max}/\hat{L} = (\max_{1\leq i\leq n}\{\|\boldsymbol{a}_i\|^2\})/(\frac{1}{n^2}\|\sum_{j=1}^n \boldsymbol{I}_{(j-1)\uparrow}\boldsymbol{A}_{\bar{\pi}}\boldsymbol{A}_{\bar{\pi}}^\top \boldsymbol{I}_{(j-1)\uparrow}\|_2)$. Since the scale of the smoothness constant of the loss functions is irrelevant for the ratio $L_{\max}/\hat{L}$ in this case, for simplicity, we take it to equal one. Note that assuming different smoothness constants over component loss functions would only make our bound better compared to related work (see Eq. (2.14) and the discussion following it).

We also compare $\hat{L}$ and $L_{\max}$ on a number of benchmarking datasets from LIBSVM [CL11], MNIST [Den12], CIFAR10 [KH$^+$09], and Broad Bioimage Benchmark Collection [LSC12]. For each dataset, we generate a uniformly random permutation π for the data matrix $\boldsymbol{A}$ and compute $\hat{L}_\pi$. We repeat this procedure 1000 times for all datasets and display the average $L_{\max}/\hat{L}_\pi$ in Table 2.2, except for `e2006train`, `CIFAR10`, `MNIST`, and `BBBC005` where we do 20 repetitions due to limitations of computation resources required for each calculation. We observe that among the datasets that we

[1] This is because it is possible for inequalities (i) and (ii) to be loose, in addition to (iii).

Table 2.2: The following table shows the computed values of $L_{\max}/\hat{L}$ where $\hat{L}$ is the empirical mean of $\hat{L}_\pi$ over random permutations. We note that the quantity $\sqrt{L_{\max}/\hat{L}}$ represents the improvement provided by the bound via our novel primal-dual perspective, compared to previous work.

Dataset	#Features (d)	#Datapoints (n)	$L_{\max}/\hat{L}$	$\log_n L_{\max}/\hat{L}$	$\log_{\min(d,n)} L_{\max}/\hat{L}$
a1a	123	1605	5.50	0.231	0.354
a9a	123	32561	5.49	0.164	0.354
BBBC005	361920	19201	18.3	0.295	0.295
BBBC010	361920	201	7.04	0.368	0.368
cifar10	3072	50000	10.0	0.213	0.287
duke	7129	44	38.0	0.962	0.962
e2006train	150360	16087	5.35	0.173	0.173
gisette	5000	6000	3.52	0.145	0.148
leu	7129	38	32.8	0.960	0.960
mnist	780	60000	19.1	0.268	0.443
news20	1355191	19996	42.1	0.378	0.378
rcv1	47236	20242	111	0.475	0.475
real-sim	20958	72309	194	0.471	0.529
sonar	60	208	6.26	0.344	0.448
tmc2007	30438	21519	10.9	0.239	0.239

consider, which contain all three data matrix "shapes" $d >> n$, $d << n$, and $d \approx n$, our novel bound dependent on $\hat{L}$ is much tighter. For instance, for `rcv1` and `real-sim` datasets, where d and n are of the same order, we observe that $L_{\max}/\hat{L}$ are approximately 111 and 194, respectively. For `news20` dataset where $d >> n$, $L_{\max}/\hat{L} \approx 42.1$. For `MNIST`, where $d << n$, $L_{\max}/\hat{L} \approx 19.1$. Further results are provided in Subsection 2.5.2.

Finally, as a justification for using the empirical mean of $\hat{L}_\pi$ over random permutations π in the results displayed in Table 2.2, we observe in our evaluations that the values of $L_{\max}/\hat{L}_\pi$ are fairly concentrated around their empirical mean values. Histogram plots showing the empirical distributions of $L_{\max}/\hat{L}_\pi$ for each of the datasets are provided in Subsection 2.5.2.

We conclude with a few additional remarks. Our results indicate that the structure of the data is important for predicting behavior of popular machine learning methods such as variants of shuffled SGD considered in our work, and thus should be incorporated in their study: as demonstrated in the Gaussian data example, considering simple finite sum structure and ignoring the dependence on the

data can lead to overly pessimistic bounds. Thus it would be interesting to provide a further theoretical study of shuffled SGD that incorporates distributional assumptions for the data. Additionally, as mentioned in the previous paragraph, we empirically observed that permutation-dependent parameter $\hat{L}_\pi$ concentrates around its mean for permutations generated uniformly at random. Thus, it would be interesting to consider whether our theoretical results can be strengthened to depend on the mean value of $\hat{L}_\pi$ (as opposed to maximum). We leave such considerations for future work.

2.5.2 Experiment Details

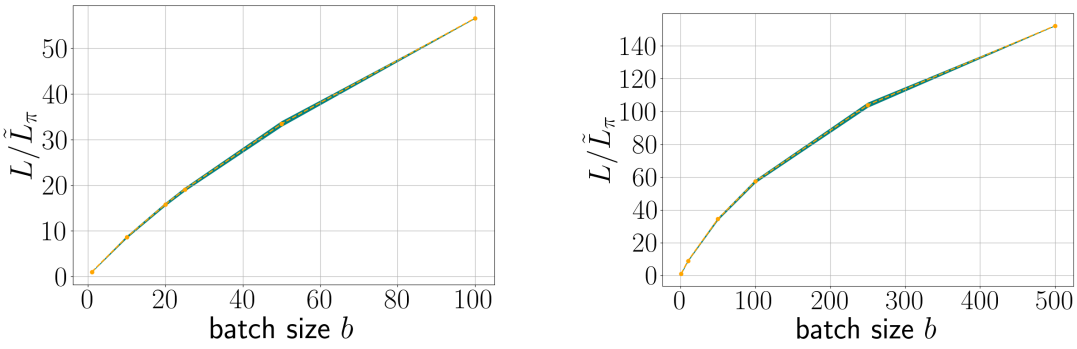
We implement the computation of $\hat{L}$ and $L_{\max}$ in Julia, a high-performance scientific computation programming language, and compute matrix operator norms using the default settings in the Julia `Arpack` Package. However, limited by computational memory and time constraint, our selection of datasets is focused on moderately large-scale datasets of n in the order of $O(10^5)$. We also include comparisons of small datasets such as `a1a` and `sonar`.

2.5.3 Evaluations of $L_{\max}/\tilde{L}_\pi$ on Synthetic Gaussian Datasets

We first study the gap between $\tilde{L}_\pi$ and $L_{\max}$ for different batch sizes b, as shown in Figure 2.2. As in Section 2.5.1, we focus on their dependence on the data matrix, and assume that the loss functions ℓ_i all have the same smoothness constant. In this case, the ratio $L_{\max}/\tilde{L}_\pi$ that characterizes the gap between $\tilde{L}_\pi$ and $L_{\max}$ will become $L_{\max}/\tilde{L}_\pi = (\max_{1\le i\le n}\{\|\boldsymbol{a}_i\|_2^2\})/(\frac{1}{b}\|\sum_{j=1}^m \boldsymbol{I}_{(j)}\boldsymbol{A}_\pi\boldsymbol{A}_\pi^\top\boldsymbol{I}_{(j)}\|_2)$. In particular, we run experiments on standard Gaussian data of size (n, d). We fix the dimension $d = 500$, and vary the number of samples with $n = 100, 500, 1000, 2000$. In Figure 2.2, we plot the ratio $L_{\max}/\tilde{L}_\pi$ versus the batch size b for 100 different random permutations π, where the dotted lines represent the mean values and the filled regions indicate the standard deviation of permutations. We observe that the ratio $L_{\max}/\tilde{L}_\pi$ is concentrated around its empirical mean and exhibits b^α ($\alpha \in [0.74, 0.87]$) growth as the batch size b increases. In particular, if we choose $b = \sqrt{n}$, the ratio can be $\mathcal{O}(n^{0.4})$.

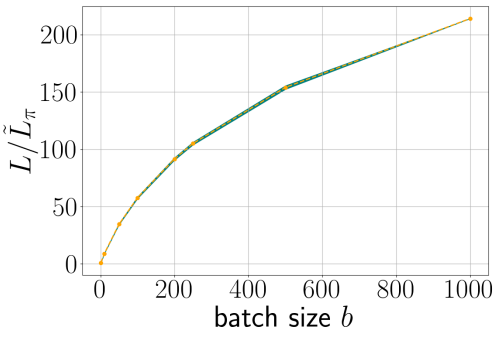
2.5.4 Distributions of $L_{\max}/\hat{L}_\pi$

In this subsection, we include histograms in Figure 2.3 to illustrate the spread of $L_{\max}/\hat{L}_\pi$ with respect to random permutations, for completeness. We observe that in all the examples $L_{\max}/\hat{L}_\pi$ is concentrated around its empirical mean. The following plots are normalized, with y-axis representing the empirical probability density. The x-axis represents $L_{\max}/\hat{L}_\pi$.

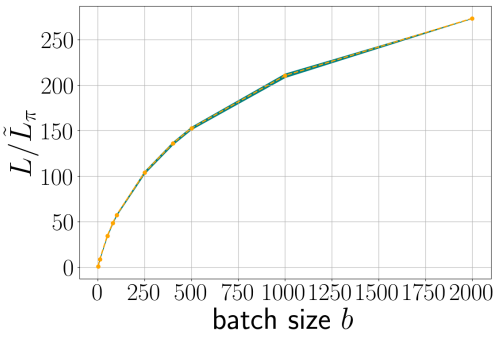


(a) $n = 100$

(b) $n = 500$



(c) $n = 1000$



(d) $n = 2000$

Figure 2.2: Illustrations of $L_{\max}/\tilde{L}_\pi$ for different batch size b on synthetic Gaussian data of size (n, d).

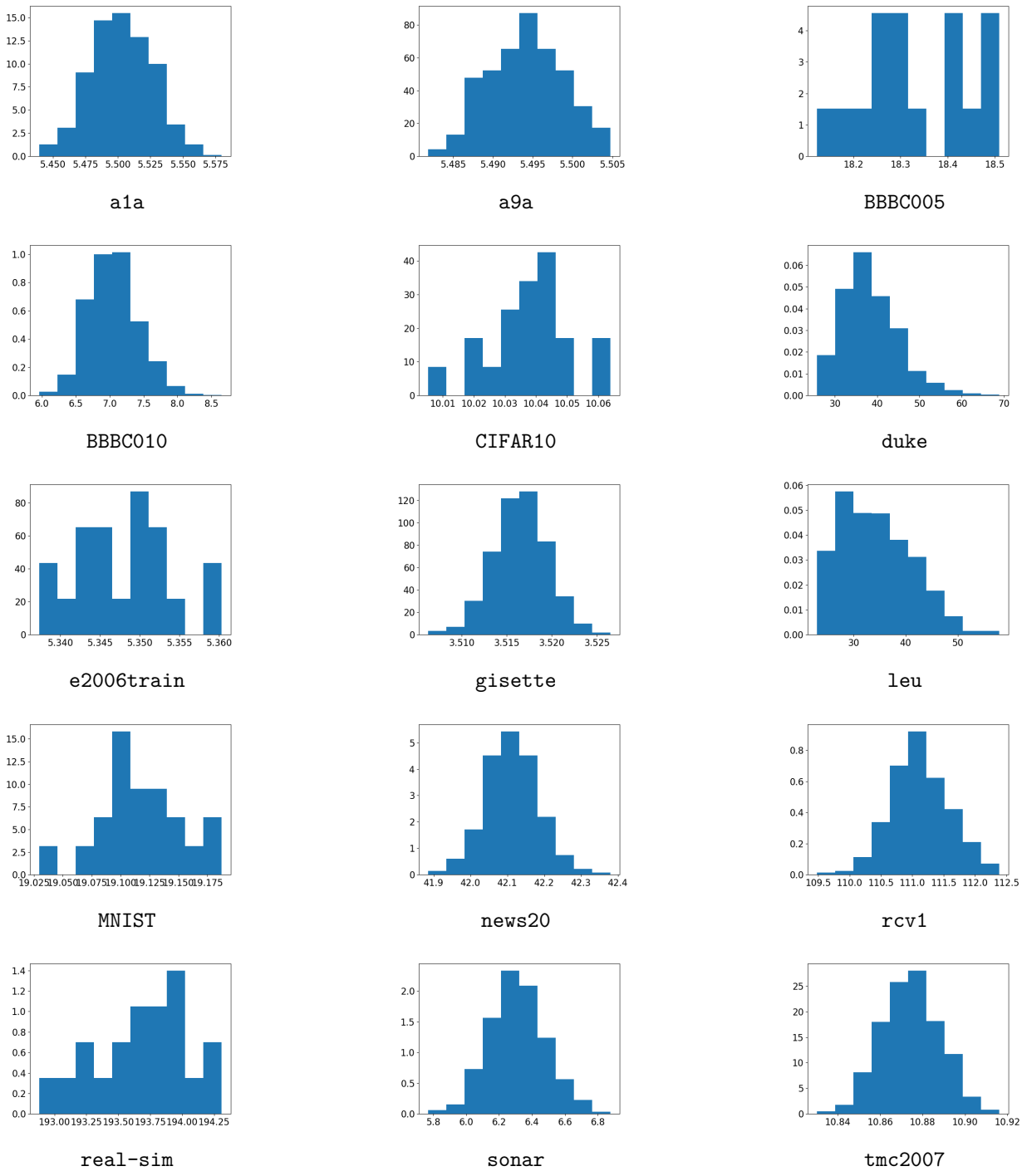


Figure 2.3: Visualization of the empirical distributions of $L/\hat{L}$ for 15 large-scale datasets.

Chapter 3

Accelerating Cyclic Coordinate Algorithms via Dual Averaging with Extrapolation

3.1 Contributions

We study the following composite convex problem

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

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, [BT09, Nes07b] 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 3.2 for a precise statement of the assumptions).

Similar to [SD21a], 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 [SD21a] (although not exactly the same) and enjoys the same favorable properties as the condition introduced in that paper; see Section 3.2 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 $O\Big(\min\Big\{ \sqrt{\frac{L}{\epsilon}} \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2,\ \sqrt{\frac{L}{\gamma}} \log(\frac{L\|\boldsymbol{x}_0-\boldsymbol{x}^*\|_2}{\epsilon}) \Big\} \Big)$, where γ is the strong convexity parameter of g (equal to zero if g is only convex, by convention), $\boldsymbol{x}^*$ is an optimal solution to (P), and $\boldsymbol{x}_0 \in \mathrm{dom}(g)$ is an arbitrary initial point. This complexity result matches the gradient oracle complexity of the fast gradient method [Nes83], 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 [BT13]. 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 [BT13]. 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 3.1, Table 3.1, and the related discussion in Section 3.2).

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 14, which shows that our Lipschitz condition translates into inequalities of the form

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

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 $f(\boldsymbol{x}) = \frac{1}{n}\sum_{t=1}^{n} f_t(\boldsymbol{x})$, 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 $O\Big(\min\Big\{\sqrt{\frac{L}{n\epsilon}}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2,\ \sqrt{\frac{L}{n\gamma}}\log(\frac{L\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2}{\epsilon})\Big\}\Big)$. The variance reduction that we employ is of the SVRG type [JZ13]. 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 14 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 [SD21a], 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.

3.1.1 Background

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 [Wri15, Nes12]. They are frequently applied to problems such as feature selection [WL$^+$08, FHT10, MFH11], empirical risk minimization [Nes12, ZL15, LLX15, AZQRY16, ADFC17, GOPV17, DO18], and in distributed computing [LWR$^+$14, FR15, RT16]. 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 [CERS18, ADFC17, AFC20, SJM20, SLWD22].

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 [Nes12, Wri15, CERS18, ADFC17, AFC20, SJM20, SLWD22, ZL15, LLX15, AZQRY16, DO18]. 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 [BT13, CWY17, SY19] and are also part of standard software packages for high-dimensional computational statistics such as GLMNet [FHT10] and SparseNet [MFH11]. 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 [BT13], 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 [BT13] scaled linearly and as a square root for vanilla CBCM and its accelerated variant, respectively. Such a high dependence on 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 [Nes83], 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 [SY19], 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 [WL20, LW19, GOPV17].

On the other hand, a very recent work in [SD21a] 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 [SD21a] 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.

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) [SY19, BT13]. Since the initial results providing such an upper bound [SY19], 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 [SD21a], 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 [JZ13, DBLJ14, AZ17, RHS$^+$16, LJCJ17, SJM20, SLRB17] for the case of full-gradient methods and in [CG16, LS18] 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 [SD21a] and nonconvex optimization [CSWD22a, XY14]. We are not aware of any existing variance reduced results for accelerated cyclic block coordinate methods.

3.1.2 Outline of the Chapter

Section 3.2 introduces the necessary notation and background and outlines our main problem assumptions. Section 3.3 introduces the A-CODER algorithm and provides the analysis. Section 3.4 presents VR-A-CODER and detailed its convergence analysis and comparison against the vanilla A-CODER. Finally, Section 3.5 provides numerical experiments for our results and concludes the paper with a discussion.

3.2 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 $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$ 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 $\mathcal{S}^j$, $j \in \{1, \ldots, m\}$, where $|\mathcal{S}^j| = d^j > 0$. For convenience of notation, we assume that sets $\mathcal{S}^j$ are comprised of consecutive elements from $\{1, 2, \ldots, d\}$, that

is, $\mathcal{S}^1 = \{1, 2, \ldots, d^1\}$, $\mathcal{S}^2 = \{d^1 + 1, d^1 + 2, \ldots, d^1 + d^2\}, \ldots, \mathcal{S}^m = \{\sum_{j=1}^{m-1} d^j + 1, \sum_{j=1}^{m-1} d^j + 2, \ldots, \sum_{j=1}^{m} d^j\}$. 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 7). For a vector $\boldsymbol{x} \in \mathrm{r}^d$, we use $\boldsymbol{x}^{(j)}$ to denote its coordinate components indexed by $\mathcal{S}^j$. 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 $\mathcal{S}^j$. We use $(\,\cdot\,)_{\geq j}$ 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 $\partial g(\boldsymbol{x})$ 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

$$\mathrm{prox}_{\tau g}(\boldsymbol{u}) := \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} \left\{ \tau g(\boldsymbol{x}) + \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{u}\|^2 \right\} \tag{3.1}$$

is efficiently computable for all $\tau > 0$ and $\boldsymbol{u} \in \mathbb{R}^d$. 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 $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and $g'(\boldsymbol{x}) \in \partial g(\boldsymbol{x})$,

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

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

Assumption 6. *$g(\boldsymbol{x})$ is γ-strongly convex, where $\gamma \geq 0$, and block-separable over coordinate sets $\{\mathcal{S}^j\}_{j=1}^m : g(\boldsymbol{x}) = \sum_{j=1}^m g^j(\boldsymbol{x}^{(j)})$. Each $g^j(\boldsymbol{x}^{(j)})$ for $j \in [m]$ admits an efficiently computable proximal operator.*

Assumption 7. *There exist positive semidefinite matrices $\{\boldsymbol{Q}^1, \boldsymbol{Q}^2, \ldots, \boldsymbol{Q}^m\}$ such that $\nabla^{(j)} f(\cdot)$ is 1-Lipschitz continuous w.r.t. the seminorm $\|\cdot\|_{\boldsymbol{Q}^j}$, i.e., $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$,*

$$\|\nabla^{(j)} f(\boldsymbol{x}) - \nabla^{(j)} f(\boldsymbol{y})\|^2 \leq \|\boldsymbol{x} - \boldsymbol{y}\|^2_{\boldsymbol{Q}^j}, \tag{3.2}$$

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

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 7 can be trivially satisfied using $\boldsymbol{Q}^j = M\boldsymbol{I}$ for all $j \in [m]$, where $\boldsymbol{I}$ is the identity matrix. Consequently, it can be argued that $L \leq 2\sqrt{m}M$ [SD21a]; however, we show that this bound is much tighter in practice as illustrated in Figure 3.1 and in Table 3.1. In particular, we follow the experiments in [SD21a] 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.

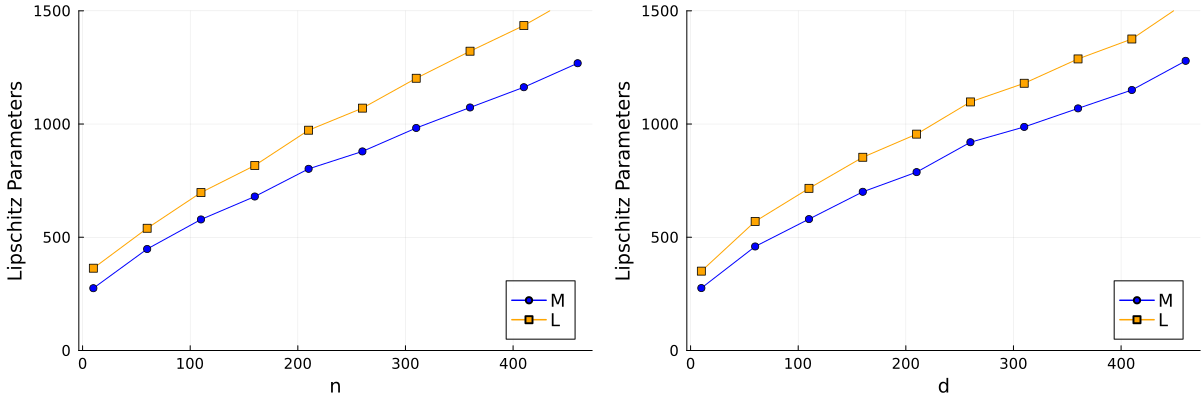


Figure 3.1: Comparisons of Lipschitz constants for elastic-net problems on synthetic datasets, where M denotes the commonly known Lipschitz constant and L is our new Lipschitz constant as defined in Assumption 7.

Table 3.1: 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

3.3 Accelerated Cyclic Algorithm

In this section, we introduce and analyze A-CODER, whose pseudocode is provided in Algorithm 3. A-CODER can be seen as a Nesterov-style accelerated variant of CODER, previously introduced for solving variational inequalities by [SD21a]. 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 $\boldsymbol{q}_k = \boldsymbol{p}_k$), A-CODER reduces to a generalized variant of AGD+ [CDO18, DG21] or the method of similar triangles [GN18]. The analysis of A-CODER follows the general gap bounding argument [DO19, SJM21] 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 7, 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 14.

Lemma 14. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex and smooth function whose gradients satisfy Assumption 7. Then, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$:*

$$\begin{aligned} f(\boldsymbol{y}) - f(\boldsymbol{x}) \leq\ & \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2, \\ \|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x})\|^2 \leq\ & 2L(f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle). \end{aligned}$$

Proof. Let $\boldsymbol{z}_j = (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(j)}, \boldsymbol{y}^{(j+1)}, \ldots, \boldsymbol{y}^{(m)})$ and observe that $\boldsymbol{z}_m = \boldsymbol{x}$ and $\boldsymbol{z}_0 = \boldsymbol{y}$. Then we have

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) = \sum_{j=1}^{m} (f(\boldsymbol{z}_{j-1}) - f(\boldsymbol{z}_j)). \tag{3.3}$$

As f is continuously differentiable and $\boldsymbol{z}_j$ and $\boldsymbol{z}_{j-1}$ only differ over the j^{th} block, we further have, by Taylor's theorem,

$$\begin{aligned} f(\boldsymbol{z}_{j-1}) - f(\boldsymbol{z}_j) &= \int_0^1 \langle \nabla f(\boldsymbol{z}_j + t(\boldsymbol{z}_{j-1} - \boldsymbol{z}_j)), \boldsymbol{z}_{j-1} - \boldsymbol{z}_j \rangle \, \mathrm{d}t \\ &= \int_0^1 \left\langle \nabla^{(j)} f(\boldsymbol{z}_j + t(\boldsymbol{z}_{j-1} - \boldsymbol{z}_j)), \boldsymbol{y}^{(j)} - \boldsymbol{x}^{(j)} \right\rangle \mathrm{d}t \\ &= \left\langle \nabla^{(j)} f(\boldsymbol{x}), \boldsymbol{y}^{(j)} - \boldsymbol{x}^{(j)} \right\rangle + \int_0^1 \left\langle \nabla^{(j)} f(\boldsymbol{z}_j + t(\boldsymbol{z}_{j-1} - \boldsymbol{z}_j)) - \nabla^{(j)} f(\boldsymbol{x}), \boldsymbol{y}^{(j)} - \boldsymbol{x}^{(j)} \right\rangle \mathrm{d}t \end{aligned} \tag{3.4}$$

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

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

where the second inequality is by our block Lipschitz assumption from Assumption 7 and the last line is by Jensen's inequality. Now observe that $\boldsymbol{z}_j$ and $\boldsymbol{x}$ agree on the first j blocks. Thus, we can write $\boldsymbol{z}_j - \boldsymbol{x} = (\boldsymbol{y} - \boldsymbol{x})_{\geq j+1}$ and $\boldsymbol{z}_{j-1} - \boldsymbol{x} = (\boldsymbol{y} - \boldsymbol{x})_{\geq j}$, while noting that we have $\|(\boldsymbol{y} - \boldsymbol{x})_{\geq j}\|_{\boldsymbol{Q}^j}^2 = \|\boldsymbol{y} - \boldsymbol{x}\|_{(\boldsymbol{Q}^j)_{\geq j}}^2$ and $\|(\boldsymbol{y} - \boldsymbol{x})_{\geq j+1}\|_{\boldsymbol{Q}^j}^2 = \|\boldsymbol{y} - \boldsymbol{x}\|_{(\boldsymbol{Q}^j)_{\geq j+1}}^2$. So by combining with Eq. (3.4) and integrating over t, we have, $\forall \alpha > 0$,

$$\begin{aligned} f(\boldsymbol{z}_{j-1}) - f(\boldsymbol{z}_j) \leq\ & \left\langle \nabla^{(j)} f(\boldsymbol{x}), \boldsymbol{y}^{(j)} - \boldsymbol{x}^{(j)} \right\rangle + \frac{1}{2\alpha}\|\boldsymbol{y}^{(j)} - \boldsymbol{x}^{(j)}\|^2 \\ &+ \frac{\alpha}{4}\Big(\|\boldsymbol{y} - \boldsymbol{x}\|_{(\boldsymbol{Q}^j)_{\geq j}}^2 + \|\boldsymbol{y} - \boldsymbol{x}\|_{(\boldsymbol{Q}^j)_{\geq j+1}}^2\Big). \end{aligned} \tag{3.5}$$

Summing Eq. (3.5) over $j \in [m]$ and using the definition of Mahalanobis norm, we finally get

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

where we used Holder's inequality and the definition of L in Assumption 7. 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 $\boldsymbol{x}, \boldsymbol{y}$ be any two points from $\mathbb{R}^d$. Define $h_{\boldsymbol{x}}(\boldsymbol{y}) = f(\boldsymbol{y}) - \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} \rangle$. Observe that $h_{\boldsymbol{x}}(\boldsymbol{y})$ is convex (as the sum of a convex function $f(\boldsymbol{y})$ and a linear function $-\langle \nabla f(\boldsymbol{x}), \boldsymbol{y} \rangle$) and is minimized at $\boldsymbol{y} = \boldsymbol{x}$ (as for any $\boldsymbol{y} \in \mathbb{R}^d$, $h_{\boldsymbol{x}}(\boldsymbol{y}) - h_{\boldsymbol{x}}(\boldsymbol{x}) = f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \geq 0$, by convexity of f). Observe further that for any $\boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^d$, we have

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

where the last inequality is by the first part of the proof. The last inequality and the fact that $\boldsymbol{x}$ minimizes $h_{\boldsymbol{x}}$ now allow us to conclude that

$$\begin{aligned} h_{\boldsymbol{x}}(\boldsymbol{x}) &\leq h_{\boldsymbol{x}}\Big(\boldsymbol{y} - \frac{1}{L} \nabla h_{\boldsymbol{x}}(\boldsymbol{y})\Big) \\ &\leq h_{\boldsymbol{x}}(\boldsymbol{y}) - \frac{1}{2L} \|\nabla h_{\boldsymbol{x}}(\boldsymbol{y})\|^2. \end{aligned}$$

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

We now derive the A-CODER algorithm. We define $\{a_k\}_{k \geq 1}$ and $\{A_k\}_{k \geq 1}$ to be sequences of positive numbers with $A_k = \sum_{i=1}^{k} a_i, a_0 = A_0 = 0$. Let $\{\boldsymbol{x}_k\}_{k \geq 0}$ be an arbitrary sequence of points in $\mathrm{dom}(g)$. Our goal here is to bound the function value gap $\bar{f}(\boldsymbol{y}_k) - \bar{f}(\boldsymbol{u})$ above for all $\boldsymbol{u} \in \mathrm{dom}(g)$. Towards this goal, we define an estimation sequence ψ_k recursively by $\psi_0(\boldsymbol{u}) = \frac{1}{2} \|\boldsymbol{u} - \boldsymbol{x}_0\|^2$ and

$$\psi_k(\boldsymbol{u}) := \psi_{k-1}(\boldsymbol{u}) + a_k (f(\boldsymbol{x}_k) + \langle \boldsymbol{q}_k, \boldsymbol{u} - \boldsymbol{x}_k \rangle + g(\boldsymbol{u}))$$

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

Lemma 15. *For any $\boldsymbol{u} \in \mathbb{R}^d$ and any sequence of vectors $\{\boldsymbol{q}_i\}_{i\geq 1}$, we have*

$$A_k(\bar{f}(\boldsymbol{y}_k) - \bar{f}(\boldsymbol{u})) \leq \sum_{i=1}^{k} E_i(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{x}_0\|^2 - \frac{1 + A_k\gamma}{2}\|\boldsymbol{u} - \boldsymbol{v}_k\|^2, \tag{3.6}$$

where

$$\begin{aligned} E_i(\boldsymbol{u}) = {} & A_i(f(\boldsymbol{y}_i) - f(\boldsymbol{x}_i)) - A_{i-1}(f(\boldsymbol{y}_{i-1}) - f(\boldsymbol{x}_i)) \\ & - a_i \langle \boldsymbol{q}_i, \boldsymbol{v}_i - \boldsymbol{x}_i \rangle + a_i \langle \nabla f(\boldsymbol{x}_i) - \boldsymbol{q}_i, \boldsymbol{x}_i - \boldsymbol{u} \rangle \\ & - \frac{1 + A_{i-1}\gamma}{2}\|\boldsymbol{v}_i - \boldsymbol{v}_{i-1}\|^2. \end{aligned} \tag{3.7}$$

Proof. As $\boldsymbol{y}_k = \frac{1}{A_k}\sum_{i=1}^{k} a_i \boldsymbol{v}_i$ and g is convex, we have $g(\boldsymbol{y}_k) \leq \frac{1}{A_k}\sum_{i=1}^{k} a_i g(\boldsymbol{v}_i)$ and thus,

$$\begin{aligned} A_k\bar{f}(\boldsymbol{y}_k) &\leq A_k f(\boldsymbol{y}_k) + \sum_{i=1}^{k} a_i g(\boldsymbol{v}_i) \\ &= \sum_{i=1}^{k}(A_i f(\boldsymbol{y}_i) - A_{i-1} f(\boldsymbol{y}_{i-1})) + \sum_{i=1}^{k} a_i g(\boldsymbol{v}_i), \end{aligned} \tag{3.8}$$

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

$$\begin{aligned} A_k\bar{f}(\boldsymbol{u}) = {} & \sum_{i=1}^{k} a_i \bar{f}(\boldsymbol{u}) \geq \sum_{i=1}^{k} a_i (f(\boldsymbol{x}_i) + \langle \nabla f(\boldsymbol{x}_i), \boldsymbol{u} - \boldsymbol{x}_i \rangle + g(\boldsymbol{u})) \\ = {} & \sum_{i=1}^{k} a_i (f(\boldsymbol{x}_i) + \langle \boldsymbol{q}_i, \boldsymbol{u} - \boldsymbol{x}_i \rangle + g(\boldsymbol{u})) + \sum_{i=1}^{k} a_i \langle \nabla f(\boldsymbol{x}_i) - \boldsymbol{q}_i, \boldsymbol{u} - \boldsymbol{x}_i \rangle \\ = {} & \psi_k(\boldsymbol{u}) - \psi_0(\boldsymbol{u}) + \sum_{i=1}^{k} a_i \langle \nabla f(\boldsymbol{x}_i) - \boldsymbol{q}_i, \boldsymbol{u} - \boldsymbol{x}_i \rangle \\ \geq {} & \psi_k(\boldsymbol{v}_k) + \frac{1 + A_k\gamma}{2}\|\boldsymbol{u} - \boldsymbol{v}_k\|^2 - \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{x}_0\|^2 \\ & + \sum_{i=1}^{k} a_i \langle \nabla f(\boldsymbol{x}_i) - \boldsymbol{q}_i, \boldsymbol{u} - \boldsymbol{x}_i \rangle, \end{aligned}$$

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

Then as $\psi_0(\boldsymbol{v}_0) = 0$, using the recursive definition of ψ_k, we have

$$\begin{aligned} \psi_k(\boldsymbol{v}_k) &= \sum_{i=1}^{k}(\psi_i(\boldsymbol{v}_i) - \psi_{i-1}(\boldsymbol{v}_{i-1})) \\ &= \sum_{i=1}^{k}\Big((\psi_{i-1}(\boldsymbol{v}_i) - \psi_{i-1}(\boldsymbol{v}_{i-1})) + a_i(f(\boldsymbol{x}_i) + \langle \boldsymbol{q}_i, \boldsymbol{v}_i - \boldsymbol{x}_i \rangle + g(\boldsymbol{v}_i))\Big) \\ &\geq \sum_{i=1}^{k}\Big(\frac{1 + A_{i-1}\gamma}{2}\|\boldsymbol{v}_i - \boldsymbol{v}_{i-1}\|^2 + a_i(f(\boldsymbol{x}_i) + \langle \boldsymbol{q}_i, \boldsymbol{v}_i - \boldsymbol{x}_i \rangle + g(\boldsymbol{v}_i))\Big), \end{aligned} \tag{3.9}$$

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

Input: $\boldsymbol{x}_0 \in \mathrm{dom}(g)$, $\gamma \geq 0$, $L > 0$, m, $\{S^1, \ldots, S^m\}$
Initialization: $\boldsymbol{x}_{-1} = \boldsymbol{x}_0 = \boldsymbol{v}_{-1} = \boldsymbol{v}_0 = \boldsymbol{y}_0$; $\boldsymbol{p}_0 = \nabla f(\boldsymbol{x}_0)$; $\boldsymbol{z}_0 = \boldsymbol{0}$; $a_0 = A_0 = 0$
for $k = 1$ to K **do**
  Set $a_k > 0$ be largest value s.t. $\frac{a_k^2}{A_k} \leq \frac{2(1+A_{k-1}\gamma)}{5L}$ where $A_k = A_{k-1} + a_k$
  $\boldsymbol{x}_k = \frac{A_{k-1}}{A_k}\boldsymbol{y}_{k-1} + \frac{a_k}{A_k}\boldsymbol{v}_{k-1}$
  for $j = m$ to 1 **do**
    $\boldsymbol{p}_k^{(j)} = \nabla^{(j)} f(\boldsymbol{x}_k^{(1)}, \ldots, \boldsymbol{x}_k^{(j)}, \boldsymbol{y}_k^{(j+1)}, \ldots, \boldsymbol{y}_k^{(m)})$
    $\boldsymbol{q}_k^{(j)} = \boldsymbol{p}_k^{(j)} + \frac{a_{k-1}}{a_k}(\nabla^{(j)} f(\boldsymbol{x}_{k-1}) - \boldsymbol{p}_{k-1}^{(j)})$
    $\boldsymbol{z}_k^{(j)} = \boldsymbol{z}_{k-1}^{(j)} + a_k \boldsymbol{q}_k^{(j)}$
    $\boldsymbol{v}_k^{(j)} = \mathrm{prox}_{A_k g^j}(\boldsymbol{x}_0^{(j)} - \boldsymbol{z}_k^{(j)})$
    $\boldsymbol{y}_k^{(j)} = \frac{A_{k-1}}{A_k}\boldsymbol{y}_{k-1}^{(j)} + \frac{a_k}{A_k}\boldsymbol{v}_k^{(j)}$
  end for
end for
return $\boldsymbol{v}_K, \boldsymbol{y}_K$

where the last inequality is by the $(1 + A_{i-1}\gamma)$-strong convexity of ψ_{i-1} and the optimality of $\boldsymbol{v}_{i-1}$. Combining Eqs. (3.8) and (3.9), we have

$$A_k(\bar{f}(\boldsymbol{y}_k) - \bar{f}(\boldsymbol{u})) \leq \sum_{i=1}^{k} E_i(\boldsymbol{u}) - \frac{1 + A_k\gamma}{2}\|\boldsymbol{u} - \boldsymbol{v}_k\|^2 + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{x}_0\|^2, \tag{3.10}$$

where $E_i(\boldsymbol{u})$ is defined in (3.7). □

Lemma 15 applies to an arbitrary algorithm that satisfies its assumptions. From now on, we make the analysis specific to A-CODER (Algorithm 3). In Lemma 15, $\{E_i(\boldsymbol{u})\}$ are the error terms that we need to bound above. If $\sum_{i=1}^{k} E_i(\boldsymbol{u}) \leq \frac{1+A_k\gamma}{2}\|\boldsymbol{u} - \boldsymbol{v}_k\|^2$, then we get the desired $1/A_k$ rate. To this end, we bound each term $E_k(\boldsymbol{u})$ in Lemma 16 by using the extrapolation direction $\boldsymbol{q}_k$, the definition of $\boldsymbol{y}_k, \boldsymbol{x}_k$ and the parameter setting of a_k.

Lemma 16. *Let $\boldsymbol{x}_0 \in \mathrm{dom}(g)$ be an arbitrary initial point and consider the updates in Algorithm 3. If, for $k \geq 1$, $\frac{{a_k}^2}{A_k} \leq \frac{2(1+A_{k-1}\gamma)}{5L}$, then $\forall \boldsymbol{u}$,*

$$\begin{aligned} E_k(\boldsymbol{u}) \leq\ & a_k \langle \nabla f(\boldsymbol{x}_k) - \boldsymbol{p}_k, \boldsymbol{v}_k - \boldsymbol{u} \rangle \\ & - a_{k-1} \langle \nabla f(\boldsymbol{x}_{k-1}) - \boldsymbol{p}_{k-1}, \boldsymbol{v}_{k-1} - \boldsymbol{u} \rangle \\ & - \frac{1 + A_{k-1}\gamma}{10}\|\boldsymbol{v}_k - \boldsymbol{v}_{k-1}\|^2 \\ & + \frac{1 + A_{k-2}\gamma}{10}\|\boldsymbol{v}_{k-1} - \boldsymbol{v}_{k-2}\|^2. \end{aligned}$$

Proof. By the convexity of f, we have

$$f(\boldsymbol{y}_{k-1}) - f(\boldsymbol{x}_k) \geq \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{y}_{k-1} - \boldsymbol{x}_k \rangle .$$

Then by applying Lemma 14, we have

$$\begin{aligned} A_k(f(\boldsymbol{y}_k) - f(\boldsymbol{x}_k)) - A_{k-1}(f(\boldsymbol{y}_{k-1}) - f(\boldsymbol{x}_k)) &\leq \langle \nabla f(\boldsymbol{x}_k), A_k\boldsymbol{y}_k - A_{k-1}\boldsymbol{y}_{k-1} - a_k\boldsymbol{x}_k \rangle + \frac{A_k L}{2}\|\boldsymbol{y}_k - \boldsymbol{x}_k\|^2 \\ &= a_k \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{v}_k - \boldsymbol{x}_k \rangle + \frac{{a_k}^2 L}{2A_k}\|\boldsymbol{v}_k - \boldsymbol{v}_{k-1}\|^2, \end{aligned} \tag{3.11}$$

where we used the definitions of $\boldsymbol{y}_k$ and $\boldsymbol{x}_k$ from Algorithm 3 in the last equality. Combining Eq. (3.7) (with $i = k$) in Lemma 15 and Eq. (3.11), we have

$$E_k(\boldsymbol{u}) \leq \Big(\frac{{a_k}^2 L}{2A_k} - \frac{1 + A_{k-1}\gamma}{2}\Big)\|\boldsymbol{v}_k - \boldsymbol{v}_{k-1}\|^2 + a_k \langle \nabla f(\boldsymbol{x}_k) - \boldsymbol{q}_k, \boldsymbol{v}_k - \boldsymbol{u} \rangle . \tag{3.12}$$

Thus by rewriting the second term as the sum of inner products over the m blocks and by using the definition of $\boldsymbol{q}_k^{(j)}$ in Algorithm 3, we have

$$\begin{aligned} a_k\langle \nabla f(\boldsymbol{x}_k) - \boldsymbol{q}_k, \boldsymbol{v}_k - \boldsymbol{u} \rangle &= a_k \sum_{j=1}^{m} \Big\langle \nabla^{(j)} f(\boldsymbol{x}_k) - \boldsymbol{q}_k^{(j)}, \boldsymbol{v}_k^{(j)} - \boldsymbol{u}^{(j)} \Big\rangle \\ &= \sum_{j=1}^{m} \Big[a_k \Big\langle \nabla^{(j)} f(\boldsymbol{x}_k) - \boldsymbol{p}_k^{(j)}, \boldsymbol{v}_k^{(j)} - \boldsymbol{u}^{(j)} \Big\rangle \\ &\qquad - a_{k-1} \Big\langle \nabla^{(j)} f(\boldsymbol{x}_{k-1}) - \boldsymbol{p}_{k-1}^{(j)}, \boldsymbol{v}_{k-1}^{(j)} - \boldsymbol{u}^{(j)} \Big\rangle \Big] \\ &\quad + a_{k-1} \sum_{j=1}^{m} \Big\langle \nabla^{(j)} f(\boldsymbol{x}_{k-1}) - \boldsymbol{p}_{k-1}^{(j)}, \boldsymbol{v}_{k-1}^{(j)} - \boldsymbol{v}_k^{(j)} \Big\rangle . \end{aligned} \tag{3.13}$$

Notice that the first two inner product terms in the first line of Eq. (3.13) telescope when summed over k, therefore it remains to bound $a_{k-1}\sum_{j=1}^{m} \Big\langle \nabla^{(j)} f(\boldsymbol{x}_{k-1}) - \boldsymbol{p}_{k-1}^{(j)}, \boldsymbol{v}_k^{(j)} - \boldsymbol{v}_{k-1}^{(j)} \Big\rangle$. In particular we let $\boldsymbol{w}_{k,j} = (\boldsymbol{x}_k^1, \ldots, \boldsymbol{x}_k^j, \boldsymbol{y}_k^{j+1}, \ldots, \boldsymbol{y}_k^m)$ so that $\boldsymbol{p}_k^{(j)} = \nabla^{(j)} f(\boldsymbol{w}_{k,j})$, then we have

$$\begin{aligned} \Big\langle \nabla^{(j)} f(\boldsymbol{x}_{k-1}) - \boldsymbol{p}_{k-1}^{(j)}, \boldsymbol{v}_{k-1}^{(j)} - \boldsymbol{v}_k^{(j)} \Big\rangle &= \Big\langle \nabla^{(j)} f(\boldsymbol{x}_{k-1}) - \nabla^{(j)} f(\boldsymbol{w}_{k-1,j}), \boldsymbol{v}_{k-1}^{(j)} - \boldsymbol{v}_k^{(j)} \Big\rangle \\ &\leq \frac{\alpha}{2} \Big\| \nabla^{(j)} f(\boldsymbol{x}_{k-1}) - \nabla^{(j)} f(\boldsymbol{w}_{k-1,j}) \Big\|^2 + \frac{1}{2\alpha} \Big\| \boldsymbol{v}_{k-1}^{(j)} - \boldsymbol{v}_k^{(j)} \Big\|^2 \\ &\leq \frac{\alpha}{2} \|\boldsymbol{x}_{k-1} - \boldsymbol{w}_{k-1,j}\|_{\boldsymbol{Q}^j}^2 + \frac{1}{2\alpha} \Big\| \boldsymbol{v}_{k-1}^{(j)} - \boldsymbol{v}_k^{(j)} \Big\|^2 \end{aligned} \tag{3.14}$$

where the first inequality holds for any $\alpha > 0$ by Young's inequality and the second inequality is by Assumption 7. Notice that $\boldsymbol{x}_{k-1}$ and $\boldsymbol{w}_{k-1,j}$ agree on the first j blocks, so similar to the proof of

Lemma 14 we can write $\boldsymbol{x}_{k-1}-\boldsymbol{w}_{k-1,j}=(\boldsymbol{y}_{k-1}-\boldsymbol{x}_{k-1})_{\geq j+1}$. Therefore by applying similar arguments as Lemma 14, we get

$$\begin{aligned}\sum_{j=1}^{m}\|\boldsymbol{x}_{k-1}-\boldsymbol{w}_{k-1,j}\|_{\boldsymbol{Q}^j}^2 &= \sum_{j=1}^{m}\|\boldsymbol{y}_{k-1}-\boldsymbol{x}_{k-1}\|_{(\boldsymbol{Q}^j)_{\geq j+1}}^2 \\ &\leq \sum_{j=1}^{m}\|\boldsymbol{y}_{k-1}-\boldsymbol{x}_{k-1}\|_{(\boldsymbol{Q}^j)_{\geq j+1}}^2+\sum_{j=1}^{m}\|\boldsymbol{y}_{k-1}-\boldsymbol{x}_{k-1}\|_{(\boldsymbol{Q}^j)_{\geq j}}^2 \\ &= \|\boldsymbol{y}_{k-1}-\boldsymbol{x}_{k-1}\|_{\tilde{\boldsymbol{Q}}}^2 \\ &\leq \frac{a_{k-1}^2L^2}{2A_{k-1}^2}\|\boldsymbol{v}_{k-1}-\boldsymbol{v}_{k-2}\|^2 \end{aligned} \tag{3.15}$$

where we used the non-negativity of Mahalanobis norm w.r.t. semi-positive definite matrix in the first inequality and the definition of $\boldsymbol{x}_k$, $\boldsymbol{y}_k$ and L in the last inequality. Lastly, by combining Eqs. (3.12), (3.13), (3.14) and (3.15), we have

$$\begin{aligned}E_k(\boldsymbol{u}) \leq\ & a_k\langle\nabla f(\boldsymbol{x}_k)-\boldsymbol{p}_k,\boldsymbol{v}_k-\boldsymbol{u}\rangle-a_{k-1}\langle\nabla f(\boldsymbol{x}_{k-1})-\boldsymbol{p}_{k-1},\boldsymbol{v}_{k-1}-\boldsymbol{u}\rangle \\ &+\left(\frac{a_k^2L}{2A_k}-\frac{1+A_{k-1}\gamma}{2}+\frac{a_{k-1}}{2\alpha}\right)\|\boldsymbol{v}_k-\boldsymbol{v}_{k-1}\|^2+\left(\frac{\alpha a_{k-1}^3L^2}{4A_{k-1}^2}\right)\|\boldsymbol{v}_{k-1}-\boldsymbol{v}_{k-2}\|^2.\end{aligned}$$

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

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

Theorem 6. *Let $\boldsymbol{x}_0\in\mathrm{dom}(g)$ be an arbitrary initial point and consider the updates in Algorithm 3. Then, $\forall k\geq 1$ and any $\boldsymbol{u}\in\mathrm{dom}(g)$:*

$$\bar{f}(\boldsymbol{y}_k)-\bar{f}(\boldsymbol{u})+\frac{3(1+A_{k-1}\gamma)}{10A_k}\|\boldsymbol{u}-\boldsymbol{v}_k\|^2\leq\frac{\|\boldsymbol{u}-\boldsymbol{x}_0\|^2}{2A_k}.$$

In particular, if $\boldsymbol{x}^*=\arg\min_{\boldsymbol{x}}\bar{f}(\boldsymbol{x})$ exists, then

$$\bar{f}(\boldsymbol{y}_k)-\bar{f}(\boldsymbol{x}^*)\leq\frac{\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2}{2A_k}.$$

Further, in this case we also have:

$$\|\boldsymbol{v}_k-\boldsymbol{x}^*\|^2\leq\frac{5}{3(1+A_{k-1}\gamma)}\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2,$$

$$\|\boldsymbol{y}_k-\boldsymbol{x}^*\|^2\leq\left(\frac{5}{3A_k}\sum_{i=1}^{k}\frac{a_i}{1+A_{i-1}\gamma}\right)\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2.$$

Finally, in all the bounds we have

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

Proof. By Lemma 16, and using the fact $A_0 = a_0 = 0$ and $\boldsymbol{v}_0 = \boldsymbol{v}_{-1}$, we have

$$\sum_{i=1}^{k} E_i(\boldsymbol{u}) \leq -\frac{1 + A_{k-1}\gamma}{10}\|\boldsymbol{v}_k - \boldsymbol{v}_{k-1}\|^2 + a_k \langle \nabla f(\boldsymbol{x}_k) - \boldsymbol{p}_k, \boldsymbol{v}_k - \boldsymbol{u}\rangle . \tag{3.16}$$

Same as in the proof of Lemma 16, we can bound $a_k \langle \nabla f(\boldsymbol{x}_k) - \boldsymbol{p}_k, \boldsymbol{v}_k - \boldsymbol{u}\rangle$ using Young's inequality and the definition of smoothness for f. In particular, for any $\alpha > 0$,

$$\begin{aligned} a_k \langle \nabla f(\boldsymbol{x}_k) - \boldsymbol{p}_k, \boldsymbol{v}_k - \boldsymbol{u}\rangle &\leq a_k\Big(\frac{\alpha L^2}{4}\|\boldsymbol{y}_k - \boldsymbol{x}_k\|^2 + \frac{1}{2\alpha}\|\boldsymbol{u} - \boldsymbol{v}_k\|^2\Big) \\ &= a_k\Big(\frac{\alpha L^2 {a_k}^2}{4{A_k}^2}\|\boldsymbol{v}_k - \boldsymbol{v}_{k-1}\|^2 + \frac{1}{2\alpha}\|\boldsymbol{u} - \boldsymbol{v}_k\|^2\Big). \end{aligned}$$

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

$$a_k \langle \nabla f(\boldsymbol{x}_k) - \boldsymbol{p}_k, \boldsymbol{v}_k - \boldsymbol{u}\rangle \leq \frac{(1 + A_{k-1}\gamma)}{10}\|\boldsymbol{v}_k - \boldsymbol{v}_{k-1}\|^2 + \frac{(1 + A_{k-1}\gamma)}{5}\|\boldsymbol{u} - \boldsymbol{v}_k\|^2. \tag{3.17}$$

Then combining Lemma 15, Eq. (3.17) and Eq. (3.16) with the fact $A_{k-1} \leq A_k$, we have

$$(\bar{f}(\boldsymbol{y}_k) - \bar{f}(\boldsymbol{u})) + \frac{3(1 + A_{k-1}\gamma)}{10A_k}\|\boldsymbol{u} - \boldsymbol{v}_k\|^2 \leq \frac{1}{2A_k}\|\boldsymbol{u} - \boldsymbol{x}_0\|^2. \tag{3.18}$$

Assume now that $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \bar{f}(\boldsymbol{x})$ exists. As $\bar{f}(\boldsymbol{y}_k) - \bar{f}(\boldsymbol{x}^*) \geq 0$, Eq. (3.18) implies

$$\|\boldsymbol{v}_k - \boldsymbol{x}^*\|^2 \leq \frac{5}{3(1 + A_{k-1}\gamma)}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2. \tag{3.19}$$

Using Jensen's inequality, as $\boldsymbol{y}_k = \frac{1}{A_k}\sum_{i=1}^{k} a_i \boldsymbol{v}_i$, we also have from Eq. (3.19)

$$\|\boldsymbol{y}_k - \boldsymbol{x}^*\|^2 \leq \left(\frac{5}{3A_k}\sum_{i=1}^{k}\frac{a_i}{1 + A_{i-1}\gamma}\right)\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2.$$

Finally, recall once again that $\{a_k\}_{k\geq 1}$ is chosen so that $\frac{{a_k}^2}{A_k} = \frac{2(1+A_{k-1}\gamma)}{5L}$. 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 $\frac{a_k}{A_{k-1}} > \sqrt{\frac{2\gamma}{5L}}$, and it remains to use that $A_k = \frac{A_k}{A_{k-1}} \cdot \ldots \cdot \frac{A_2}{A_1} \cdot A_1 = A_1\Big(1 + \sqrt{\frac{2\gamma}{5L}}\Big)^{k-1}$ where $a_1 = A_1 = \frac{2}{5L}$ using the choice of a_k in Algorithm 3 and $A_0 = a_0 = 0$, completing the proof. □

3.3.1 (Lipschitz) Parameter-Free A-CODER

Adaptive A-CODER. The Lipschitz parameter L used in the statement of A-CODER (Algorithm 3) is usually not readily available for typical instances of convex composite minimization problems. However 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 3.3 to apply is

Algorithm 4 Adaptive Accelerated Cyclic cOordinate Dual avEraging with extRapolation (Ada-A-CODER)

Input: $\boldsymbol{x}_0 \in \mathrm{dom}(g), \gamma \geq 0, L_0 > 0, m, \{S^1, \ldots, S^m\}$
Initialization: $\boldsymbol{x}_{-1} = \boldsymbol{x}_0 = \boldsymbol{v}_{-1} = \boldsymbol{v}_0 = \boldsymbol{y}_0, \boldsymbol{p}_0 = \nabla f(\boldsymbol{x}_0), \boldsymbol{z}_0 = \boldsymbol{0}$, $a_0 = A_0 = 0$
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. $\frac{a_k^2}{A_k} \leq \frac{2(1+A_{k-1}\gamma)}{5L_k}$ where $A_k = A_{k-1} + a_k$
    $\boldsymbol{x}_k = \frac{A_{k-1}}{A_k}\boldsymbol{y}_{k-1} + \frac{a_k}{A_k}\boldsymbol{v}_{k-1}$
    for $j = m$ to 1 **do**
      $\boldsymbol{p}_k^{(j)} = \nabla^{(j)} f(\boldsymbol{x}_k^{(1)}, \ldots, \boldsymbol{x}_k^{(j)}, \boldsymbol{y}_k^{(j+1)}, \ldots, \boldsymbol{y}_k^{(m)})$
      $\boldsymbol{q}_k^{(j)} = \boldsymbol{p}_k^{(j)} + \frac{a_{k-1}}{a_k}(\nabla^{(j)} f(\boldsymbol{x}_{k-1}) - \boldsymbol{p}_{k-1}^{(j)})$
      $\boldsymbol{z}_k^{(j)} = \boldsymbol{z}_{k-1}^{(j)} + a_k \boldsymbol{q}_k^{(j)}$
      $\boldsymbol{v}_k^{(j)} = \mathrm{prox}_{A_k g^j}(\boldsymbol{x}_0^{(j)} - \boldsymbol{z}_k^{(j)})$
      $\boldsymbol{y}_k^{(j)} = \frac{A_{k-1}}{A_k}\boldsymbol{y}_{k-1}^{(j)} + \frac{a_k}{A_k}\boldsymbol{v}_k^{(j)}$
    end for
  until $f(\boldsymbol{y}_k) \leq f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{y}_k - \boldsymbol{x}_k \rangle + \frac{L_k}{2}\|\boldsymbol{y}_k - \boldsymbol{x}_k\|^2$
end for
return $\boldsymbol{v}_K, \boldsymbol{y}_K$

that the quadratic bound from Lemma 14 holds between $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$. A variant of A-CODER that implements this adaptive estimation is provided in Algorithm 4.

A variant of A-CODER implementing this adaptive estimation of L is provided in Algorithm 4. 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., [Nes12]), we only need to estimate one summary Lipschitz parameter L. Note that our adaptive version of A-CODER does not require extra gradient computations in each iteration to verify the smoothness condition, unlike in RCDM [Nes12].

3.4 Variance Reduced A-CODER

In this section, we assume that the problem (P) has a finite sum structure, i.e., $f(\boldsymbol{x}) = \frac{1}{n}\sum_{t=1}^{n} f_t(\boldsymbol{x})$, where n may be very large. For this case, we can further reduce the per-iteration

Algorithm 5 Variance Reduced A-CODER (Implementable Version)

Input: $\boldsymbol{x}_0 \in \mathrm{dom}(g)$, $\gamma \geq 0$, $L > 0$, m, $\{\mathcal{S}^1, \ldots, \mathcal{S}^m\}$
Initialization: $\tilde{\boldsymbol{y}}_0 = \boldsymbol{v}_{1,0} = \boldsymbol{y}_{1,0} = \boldsymbol{x}_{1,1} = \boldsymbol{x}_0$; $\boldsymbol{z}_{1,0} = \boldsymbol{0}$
$a_0 = A_0 = 0$; $A_1 = a_1 = \frac{1}{4L}$
$\boldsymbol{z}_{1,1} = \nabla f(\boldsymbol{x}_0)$; $\boldsymbol{v}_{1,1} = \mathrm{prox}_{a_1 g}(\boldsymbol{x}_0 - \boldsymbol{z}_{1,1})$
$\tilde{\boldsymbol{y}}_1 = \boldsymbol{y}_{1,1} = \boldsymbol{v}_{1,1}$
$\boldsymbol{w}_{1,1,j} = (\boldsymbol{x}_{1,1}^{(1)}, \ldots, \boldsymbol{x}_{1,1}^{(j)}, \boldsymbol{y}_{1,1}^{(j+1)}, \ldots, \boldsymbol{y}_{1,1}^{(m)})$
$\boldsymbol{v}_{2,0} = \boldsymbol{v}_{1,1}$; $\boldsymbol{w}_{2,0,j} = \boldsymbol{w}_{1,1,j}$; $\boldsymbol{x}_{2,0} = \boldsymbol{x}_{1,1}$; $\boldsymbol{y}_{2,0} = \boldsymbol{y}_{1,1}$; $\boldsymbol{z}_{2,0} = \boldsymbol{z}_{1,1}$
for $s = 2$ to S **do**
  $a_s = \sqrt{\frac{KA_{s-1}(1+A_{s-1}\gamma)}{8L}}$; $A_s = A_{s-1} + a_s$
  $a_{s,0} = a_{s-1}$; $a_{s,1} = a_{s,2} = \cdots = a_{s,K} = a_s$
  $\boldsymbol{v}_{s,0} = \boldsymbol{v}_{s-1,K}$; $\boldsymbol{w}_{s,0,j} = \boldsymbol{w}_{s-1,K,j}$; $\boldsymbol{x}_{s,0} = \boldsymbol{x}_{s-1,K}$; $\boldsymbol{y}_{s,0} = \boldsymbol{y}_{s-1,K}$; $\boldsymbol{z}_{s,0} = \boldsymbol{z}_{s-1,K}$
  $\boldsymbol{\mu}_s = \nabla f(\tilde{\boldsymbol{y}}_{s-1})$
  for $k = 1$ to K **do**
    $\boldsymbol{x}_{s,k} = \frac{A_{s-1}}{A_s}\tilde{\boldsymbol{y}}_{s-1} + \frac{a_s}{A_s}\boldsymbol{v}_{s,k-1}$
    for $j = m$ to 1 **do**
      $\boldsymbol{w}_{s,k,j} = (\boldsymbol{x}_{s,k}^{(1)}, \ldots, \boldsymbol{x}_{s,k}^{(j)}, \boldsymbol{y}_{s,k}^{(j+1)}, \ldots, \boldsymbol{y}_{s,k}^{(m)})$
      Choose t in $[n]$ uniformly at random
      $\tilde{\nabla}_{s,k}^{(j)} = \nabla^{(j)} f_t(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_t(\tilde{\boldsymbol{y}}_{s-1}) + \boldsymbol{\mu}_s^{(j)}$
      $\boldsymbol{q}_{s,k}^{(j)} = \tilde{\nabla}_{s,k}^{(j)} + \frac{a_{s,k-1}}{a_s}(\nabla^{(j)} f_t(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)} f_t(\boldsymbol{w}_{s,k-1,j}))$
      $\boldsymbol{z}_{s,k}^{(j)} = \boldsymbol{z}_{s,k-1}^{(j)} + a_s \boldsymbol{q}_{s,k}^{(j)}$
      $\boldsymbol{v}_{s,k}^{(j)} = \mathrm{prox}_{(A_{s-1} + \frac{a_s k}{K}) g^j}(\boldsymbol{x}_0^{(j)} - \boldsymbol{z}_{s,k}^{(j)}/K)$
      $\boldsymbol{y}_{s,k}^{(j)} = \frac{A_{s-1}}{A_s}\tilde{\boldsymbol{y}}_{s-1}^{(j)} + \frac{a_s}{A_s}\boldsymbol{v}_{s,k}^{(j)}$
    end for
  end for
  $\tilde{\boldsymbol{y}}_s = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}_{s,k}$
end for
return $\boldsymbol{v}_{S,K}, \tilde{\boldsymbol{y}}_S$

cost and improve the complexity results by combining the well-known SVRG-style variance reduction strategy [JZ13] 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 [SJM20].

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

Algorithm 6 Variance Reduced A-CODER (Analysis Version)

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

Assumption 8. *For all $t \in [n]$, $f_t(\boldsymbol{x})$ is convex. Moreover for all $t \in [n]$, there exist positive semi-definite matrices $\{\boldsymbol{Q}^1, \boldsymbol{Q}^2, \ldots, \boldsymbol{Q}^m\}$ such that $\nabla^{(j)} f_t(\cdot)$ is 1-Lipschitz continuous w.r.t. the norm $\|\cdot\|_{\boldsymbol{Q}^j}$ i.e., $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, t \in [n]$,*

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

Lemma 17. *If $f(\boldsymbol{x}) = \frac{1}{n}\sum_{t=1}^{n} f_t(\boldsymbol{x})$ satisfies Assumption 8, then it satisfies Assumption 7 and thus Lemma 14 holds.*

Proof. By using Jensen's inequality and Assumption 8, we have

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

□

With this assumption, we can now derive the VR-A-CODER algorithm. Similar to Section 3.3, we define $\{a_s\}_{s\geq 1}$ and $\{A_s\}_{s\geq 1}$ to be sequences of positive numbers with $A_s = \sum_{i=1}^{s} a_i, a_0 = A_0 = 0$. Let $\{\tilde{\boldsymbol{y}}_s\}_{s\geq 0}$ 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 $\bar{f}(\tilde{\boldsymbol{y}}_s) - \bar{f}(\boldsymbol{u})$ above ($\boldsymbol{u} \in \mathrm{dom}(g)$). To attain this, we define the estimate sequence $\{\psi_{s,k}\}_{s\geq 1, k\in[K]}$ recursively by $\psi_{1,0}(\boldsymbol{u}) = \frac{K}{2}\|\boldsymbol{u}-\boldsymbol{x}_0\|^2$,

$$\psi_{1,1}(\boldsymbol{u}) = \psi_{1,0}(\boldsymbol{u}) + Ka_1(f(\boldsymbol{x}_0) + \langle \nabla f(\boldsymbol{x}_0), \boldsymbol{u}-\boldsymbol{x}_0\rangle + g(\boldsymbol{u})), \tag{3.20}$$

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

$$\psi_{s,k}(\boldsymbol{u}) = \psi_{s,k-1}(\boldsymbol{u}) + a_s(f(\boldsymbol{x}_{s,k}) + \langle \boldsymbol{q}_{s,k}, \boldsymbol{u}-\boldsymbol{x}_{s,k}\rangle + g(\boldsymbol{u})), \tag{3.21}$$

and $\psi_{s+1,0} = \psi_{s,K}$. In Eqs. (3.20) and (3.21), $\boldsymbol{x}_0$ is the initial point, $\boldsymbol{x}_{s,k}$ and $\boldsymbol{y}_{s,k}$ are computed as convex combinations of two points, which is commonly used in Nesterov-style acceleration, and $\boldsymbol{q}_{s,k} = (\boldsymbol{q}_{s,k}^{(1)}, \boldsymbol{q}_{s,k}^{(2)}, \ldots, \boldsymbol{q}_{s,k}^{(m)})$ is a variance reduced stochastic gradient with extrapolation, which is the main novelty in our algorithm design. Meanwhile, we define $\boldsymbol{v}_{s,k}$ by $\boldsymbol{v}_{s,k} := \arg\min_{\boldsymbol{u}\in\mathbb{R}^d} \psi_{s,k}(\boldsymbol{u})$ and note that due to the specific choice of $\boldsymbol{q}_{s,k}$, $\boldsymbol{v}_{s,k}$ is updated in a cyclic (block) coordinate way. Furthermore, in VR-A-CODER, for $s \geq 2$, we define $\tilde{\boldsymbol{y}}_s = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}_{s,k}$.

We start our analysis by characterizing the gap function, in the following lemma, similar to Lemma 15 in Section 3.3, although the proof is much more technical in this case.

Lemma 18. *For any $\boldsymbol{u} \in \mathbb{R}^d$ and any sequence of vectors $\{\boldsymbol{q}_{s,k}\}_{s\geq 2, k\in[K]}$, for all $S \geq 2$, we have*

$$\begin{aligned} &KA_S(\bar{f}(\tilde{\boldsymbol{y}}_S) - \bar{f}(\boldsymbol{u})) \\ &\leq \frac{K}{2}\|\boldsymbol{x}_0-\boldsymbol{u}\|^2 - \frac{K(1+A_S\gamma)}{2}\|\boldsymbol{v}_{S,K}-\boldsymbol{u}\|^2 - \frac{K}{4}\|\boldsymbol{v}_{1,1}-\boldsymbol{v}_{1,0}\|^2 + \sum_{s=2}^{S}\sum_{k=1}^{K} E_{s,k}(\boldsymbol{u}), \end{aligned}$$

where

$$\begin{aligned} E_{s,k}(\boldsymbol{u}) = {} & A_s(f(\boldsymbol{y}_{s,k}) - f(\boldsymbol{x}_{s,k})) - A_{s-1}(f(\tilde{\boldsymbol{y}}_{s-1}) - f(\boldsymbol{x}_{s,k})) + a_s\langle \nabla f(\boldsymbol{x}_{s,k}) - \boldsymbol{q}_{s,k}, \boldsymbol{x}_{s,k}-\boldsymbol{u}\rangle \\ & + a_s\langle \boldsymbol{q}_{s,k}, \boldsymbol{x}_{s,k}-\boldsymbol{v}_{s,k}\rangle - \frac{K(1+A_{s-1}\gamma)}{2}\|\boldsymbol{v}_{s,k}-\boldsymbol{v}_{s,k-1}\|^2. \end{aligned} \tag{3.22}$$

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

$$\begin{aligned} KA_S\bar{f}(\boldsymbol{u}) &= \sum_{s=1}^{S}\sum_{k=1}^{K} a_s \bar{f}(\boldsymbol{u}) \\ &\geq Ka_1(f(\boldsymbol{x}_{1,1}) + \langle \nabla f(\boldsymbol{x}_{1,1}), \boldsymbol{u} - \boldsymbol{x}_{1,1}\rangle + g(\boldsymbol{u})) \\ &\quad + \sum_{s=2}^{S}\sum_{k=1}^{K} a_s(f(\boldsymbol{x}_{s,k}) + \langle \nabla f(\boldsymbol{x}_{s,k}), \boldsymbol{u} - \boldsymbol{x}_{s,k}\rangle + g(\boldsymbol{u})) \\ &= \psi_{S,K}(\boldsymbol{u}) - \psi_{1,0}(\boldsymbol{u}) + \sum_{s=2}^{S}\sum_{k=1}^{K} a_s \left\langle \nabla f(\boldsymbol{x}_{s,k}) - \boldsymbol{q}_{s,k}, \boldsymbol{u} - \boldsymbol{x}_{s,k}\right\rangle \\ &\geq \psi_{S,K}(\boldsymbol{v}_{S,K}) + \frac{K(1+A_S\gamma)}{2}\|\boldsymbol{u} - \boldsymbol{v}_{S,K}\|^2 - \frac{K}{2}\|\boldsymbol{x}_0 - \boldsymbol{u}\|^2 \\ &\quad + \sum_{s=2}^{S}\sum_{k=1}^{K} a_s \left\langle \nabla f(\boldsymbol{x}_{s,k}) - \boldsymbol{q}_{s,k}, \boldsymbol{u} - \boldsymbol{x}_{s,k}\right\rangle, \end{aligned} \tag{3.23}$$

where the first inequality is by the convexity of f, the second equality is by the recursive definition of $\psi_{S,K}(\boldsymbol{u})$, the second inequality is by the $K(1 + A_S\gamma)$-strong convexity of $\psi_{S,K}(\boldsymbol{u})$ and $\boldsymbol{v}_{S,K} = \arg\min_{\boldsymbol{u}} \psi_{S,K}(\boldsymbol{u})$ leading to $\psi_{S,K}(\boldsymbol{u}) \geq \psi_{S,K}(\boldsymbol{v}_{S,K}) + \frac{K(1+A_S\gamma)}{2}\|\boldsymbol{u} - \boldsymbol{v}_{S,K}\|^2$.

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

$$\begin{aligned} &\psi_{S,K}(\boldsymbol{v}_{S,K}) \\ &= \psi_{1,1}(\boldsymbol{v}_{1,1}) + \sum_{s=2}^{S}\sum_{k=1}^{K}(\psi_{s,k}(\boldsymbol{v}_{s,k}) - \psi_{s,k-1}(\boldsymbol{v}_{s,k-1})) \\ &= \psi_{1,1}(\boldsymbol{v}_{1,1}) + \sum_{s=2}^{S}\sum_{k=1}^{K}(\psi_{s,k-1}(\boldsymbol{v}_{s,k}) - \psi_{s,k-1}(\boldsymbol{v}_{s,k-1})) \\ &\quad + \sum_{s=2}^{S}\sum_{k=1}^{K} a_s(f(\boldsymbol{x}_{s,k}) + \langle \boldsymbol{q}_{s,k}, \boldsymbol{v}_{s,k} - \boldsymbol{x}_{s,k}\rangle + g(\boldsymbol{v}_{s,k})) \\ &\geq \frac{K}{2}\|\boldsymbol{v}_{1,1} - \boldsymbol{v}_{1,0}\|^2 + Ka_1(f(\boldsymbol{x}_{1,1}) + \langle \nabla f(\boldsymbol{x}_{1,1}), \boldsymbol{v}_{1,1} - \boldsymbol{x}_{1,1}\rangle + g(\boldsymbol{v}_{1,1})) \\ &\quad + \sum_{s=2}^{S}\sum_{k=1}^{K}\frac{K(1+A_{s-1}\gamma)}{2}\|\boldsymbol{v}_{s,k} - \boldsymbol{v}_{s,k-1}\|^2 \\ &\quad + \sum_{s=2}^{S}\sum_{k=1}^{K} a_s(f(\boldsymbol{x}_{s,k}) + \langle \boldsymbol{q}_{s,k}, \boldsymbol{v}_{s,k} - \boldsymbol{x}_{s,k}\rangle + g(\boldsymbol{v}_{s,k})), \end{aligned} \tag{3.24}$$

where the first equality is by $\psi_{s+1,0} = \psi_{s,K}$ and $\boldsymbol{v}_{s+1,0} = \boldsymbol{v}_{s,K}$, the second equality is by the definition of $\psi_{s,k}$, the last inequality is by the definition of $\psi_{1,1}(\boldsymbol{v}_{1,1})$ and the $K(1 + A_{s-1}\gamma)$-strong convexity

of $\psi_{s,k-1}(s \geq 2, k \in [K])$. Then by Lemmas 17 and 14, we have

$$\begin{aligned} f(\boldsymbol{v}_{1,1}) &\leq f(\boldsymbol{x}_{1,1}) + \langle \nabla f(\boldsymbol{x}_{1,1}), \boldsymbol{v}_{1,1} - \boldsymbol{x}_{1,1} \rangle + \frac{L}{2} \|\boldsymbol{v}_{1,1} - \boldsymbol{x}_{1,1}\|^2 \\ &\leq f(\boldsymbol{x}_{1,1}) + \langle \nabla f(\boldsymbol{x}_{1,1}), \boldsymbol{v}_{1,1} - \boldsymbol{x}_{1,1} \rangle + \frac{1}{4a_1} \|\boldsymbol{v}_{1,1} - \boldsymbol{v}_{1,0}\|^2, \end{aligned} \tag{3.25}$$

where the last inequality is by $a_1 \leq \frac{1}{4L}$ and $\boldsymbol{v}_{1,0} = \boldsymbol{x}_{1,1}$.

Using $\tilde{\boldsymbol{y}}_s = \frac{1}{K} \sum_{k=1}^K \boldsymbol{y}_{s,k} = \frac{A_{s-1}}{A_s} \tilde{\boldsymbol{y}}_{s-1} + \frac{a_s}{K A_s} \sum_{k=1}^K \boldsymbol{v}_{s,k}$, the convexity of g, and $A_0 = 0$, we have

$$\begin{aligned} \sum_{s=2}^S \sum_{k=1}^K a_s g(\boldsymbol{v}_{s,k}) &\geq \sum_{s=2}^S K a_s g\Big(\frac{1}{K} \sum_{k=1}^K \boldsymbol{v}_{s,k}\Big) \geq \sum_{s=2}^S \left(K A_s g(\tilde{\boldsymbol{y}}_s) - K A_{s-1} g(\tilde{\boldsymbol{y}}_{s-1})\right) \\ &= K A_S g(\tilde{\boldsymbol{y}}_S) - K A_1 g(\tilde{\boldsymbol{y}}_1) \\ &= K A_S g(\tilde{\boldsymbol{y}}_S) - K A_1 g(\boldsymbol{v}_{1,1}). \end{aligned} \tag{3.26}$$

Thus, combining Eqs. (3.23)–(3.26), we have

$$\begin{aligned} K A_S \bar{f}(\boldsymbol{u}) \geq{}& \frac{K(1 + A_S \gamma)}{2} \|\boldsymbol{u} - \boldsymbol{v}_{S,K}\|^2 - \frac{K}{2} \|\boldsymbol{x}_0 - \boldsymbol{u}\|^2 + \frac{K}{4} \|\boldsymbol{v}_{1,1} - \boldsymbol{v}_{1,0}\|^2 + K a_1 f(\boldsymbol{v}_{1,1}) \\ &+ \sum_{s=2}^S \sum_{k=1}^K \Big(a_s \left\langle \nabla f(\boldsymbol{x}_{s,k}) - \boldsymbol{q}_{s,k}, \boldsymbol{u} - \boldsymbol{x}_{s,k} \right\rangle + \frac{K(1 + A_{s-1}\gamma)}{2} \|\boldsymbol{v}_{s,k} - \boldsymbol{v}_{s,k-1}\|^2 \Big) \\ &+ \sum_{s=2}^S \sum_{k=1}^K a_s (f(\boldsymbol{x}_{s,k}) + \langle \boldsymbol{q}_{s,k}, \boldsymbol{v}_{s,k} - \boldsymbol{x}_{s,k} \rangle) + K A_S g(\tilde{\boldsymbol{y}}_S). \end{aligned} \tag{3.27}$$

Then with $A_0 = 0$ and $\tilde{\boldsymbol{y}}_s = \frac{1}{K} \sum_{k=1}^K \boldsymbol{y}_{s,k}$, we also have

$$\begin{aligned} K A_S f(\tilde{\boldsymbol{y}}_S) &= K A_1 f(\tilde{\boldsymbol{y}}_1) + K \sum_{s=2}^S (A_s f(\tilde{\boldsymbol{y}}_s) - A_{s-1} f(\tilde{\boldsymbol{y}}_{s-1})) \\ &\leq K a_1 f(\boldsymbol{v}_{1,1}) + \sum_{s=2}^S \sum_{k=1}^K A_s f(\boldsymbol{y}_{s,k}) - K \sum_{s=2}^S A_{s-1} f(\tilde{\boldsymbol{y}}_{s-1}), \end{aligned} \tag{3.28}$$

where the last equality is by $A_1 = a_1, \tilde{\boldsymbol{y}}_1 = \boldsymbol{v}_{1,1}$. Subtracting Eq. (3.27) from (3.28) and noting that $\bar{f}(\tilde{\boldsymbol{y}}_S) = f(\tilde{\boldsymbol{y}}_S) + g(\tilde{\boldsymbol{y}}_S)$ now completes the proof. □

Lemma 19. *The error sequence* $\{E_{s,k}(\boldsymbol{u})\}_{s \geq 2, k \in [K]}$ *in Lemma 18 satisfies*

$$\begin{aligned} E_{s,k}(\boldsymbol{u}) \leq{}& -A_{s-1}(f(\tilde{\boldsymbol{y}}_{s-1}) - f(\boldsymbol{x}_{s,k}) - \langle \nabla f(\boldsymbol{x}_{s,k}), \tilde{\boldsymbol{y}}_{s-1} - \boldsymbol{x}_{s,k} \rangle) \\ &+ a_s \langle \nabla f(\boldsymbol{x}_{s,k}) - \boldsymbol{q}_{s,k}, \boldsymbol{v}_{s,k} - \boldsymbol{u} \rangle + \Big(\frac{L {a_s}^2}{2A_s} - \frac{K(1 + A_{s-1}\gamma)}{2} \Big) \|\boldsymbol{v}_{s,k} - \boldsymbol{v}_{s,k-1}\|^2. \end{aligned}$$

Proof. Using Assumption 8, Lemma 17, and Lemma 14, and applying the definition of $\boldsymbol{y}_{s,k}$, we have

$$\begin{aligned}
&f(\boldsymbol{y}_{s,k}) - f(\boldsymbol{x}_{s,k}) \\
&\leq \langle \nabla f(\boldsymbol{x}_{s,k}), \boldsymbol{y}_{s,k} - \boldsymbol{x}_{s,k} \rangle + \frac{L}{2}\|\boldsymbol{y}_{s,k} - \boldsymbol{x}_{s,k}\|^2 \\
&= \langle \nabla f(\boldsymbol{x}_{s,k}), \frac{A_{s-1}}{A_s}\tilde{\boldsymbol{y}}_{s-1} + \frac{a_s}{A_s}\boldsymbol{v}_{k,s} - \boldsymbol{x}_{s,k} \rangle + \frac{La_s{}^2}{2A_s{}^2}\|\boldsymbol{v}_{s,k} - \boldsymbol{v}_{s,k-1}\|^2 \\
&= \frac{A_{s-1}}{A_s}\langle \nabla f(\boldsymbol{x}_{s,k}), \tilde{\boldsymbol{y}}_{s-1} - \boldsymbol{x}_{s,k} \rangle + \frac{a_s}{A_s}\langle \nabla f(\boldsymbol{x}_{s,k}), \boldsymbol{v}_{s,k} - \boldsymbol{x}_{s,k} \rangle + \frac{La_s{}^2}{2A_s{}^2}\|\boldsymbol{v}_{s,k} - \boldsymbol{v}_{s,k-1}\|^2. \quad (3.29)
\end{aligned}$$

It remains to plug Eq. (3.29) into the definition of $E_{s,k}(\boldsymbol{u})$, and rearrange. □

The definition of the variance reduced extrapolation point $\boldsymbol{q}_{s,k}$ is crucial for bounding the error terms $\{E_{s,k}(\boldsymbol{u})\}$ from Lemma 18. The next three auxiliary lemmas apply the definition of $\boldsymbol{q}_{s,k}^{(j)}$ to bound the inner product term $\langle \nabla f(\boldsymbol{x}_{s,k}) - \boldsymbol{q}_{s,k}, \boldsymbol{v}_{s,k} - \boldsymbol{u} \rangle$ in $E_{s,k}(\boldsymbol{u})$ when we take the expectation over all randomness in the algorithm. We will use $\mathcal{F}_{s,k,i}$ 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 6, 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 $j = m - i + 1$. This detail however does not play a important role in our analysis.

Lemma 20. *For all $s \geq 2$, $k \in [K]$ and $\boldsymbol{u} \in \mathrm{dom}(g)$, we have*

$$\begin{aligned}
&a_s\mathbb{E}[\langle \nabla f(\boldsymbol{x}_{s,k}) - \boldsymbol{q}_{s,k}, \boldsymbol{v}_{s,k} - \boldsymbol{u} \rangle] \\
&= \sum_{j=1}^{m} a_s\mathbb{E}[\langle \nabla^{(j)} f(\boldsymbol{x}_{s,k}) - \nabla^{(j)} f(\boldsymbol{w}_{s,k,j}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{u}^{(j)} \rangle] \\
&\quad - \sum_{j=1}^{m} a_{s,k-1}\mathbb{E}[\langle \nabla^{(j)} f(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)} f(\boldsymbol{w}_{s,k-1,j}), \boldsymbol{v}_{s,k-1}^{(j)} - \boldsymbol{u}^{(j)} \rangle] \\
&\quad - \sum_{j=1}^{m} a_{s,k-1}\mathbb{E}[\langle \nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k-1,j}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)} \rangle] \\
&\quad + \sum_{j=1}^{m} a_s\mathbb{E}[\langle \nabla^{(j)} f(\boldsymbol{w}_{s,k,j}) - (\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1}) + \boldsymbol{\mu}_s^{(j)}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)} \rangle],
\end{aligned}$$

Proof. Using the definition of $\boldsymbol{q}_{s,k}^{(j)}$, we have

$$\begin{aligned}
&a_s(\nabla^{(j)} f(\boldsymbol{x}_{s,k}) - \boldsymbol{q}_{s,k}^{(j)}) \\
&= a_s(\nabla^{(j)} f(\boldsymbol{x}_{s,k}) - \nabla^{(j)} f(\boldsymbol{w}_{s,k,j})) + a_s(\nabla^{(j)} f(\boldsymbol{w}_{s,k,j}) - \boldsymbol{q}_{s,k}^{(j)}) \\
&= a_s(\nabla^{(j)} f(\boldsymbol{x}_{s,k}) - \nabla^{(j)} f(\boldsymbol{w}_{s,k,j})) + a_s(\nabla^{(j)} f(\boldsymbol{w}_{s,k,j}) - (\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1}) + \boldsymbol{\mu}_s^{(j)})) \\
&\quad - a_{s,k-1}(\nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k-1,j})). \quad (3.30)
\end{aligned}$$

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

$$\begin{aligned}
&\mathbb{E}[a_s\langle\nabla^{(j)} f(\boldsymbol{w}_{s,k,j}) - (\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1}) + \boldsymbol{\mu}_s^{(j)}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{u}^{(j)}\rangle] \\
&= \mathbb{E}[a_s\langle\nabla^{(j)} f(\boldsymbol{w}_{s,k,j}) - (\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1}) + \boldsymbol{\mu}_s^{(j)}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)}\rangle] \\
&\quad + a_s\mathbb{E}[\langle\mathbb{E}[\nabla^{(j)} f(\boldsymbol{w}_{s,k,j}) - (\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1}) + \boldsymbol{\mu}_s^{(j)})|\mathcal{F}_{s,k,j-1}], \boldsymbol{v}_{s,k-1}^{(j)} - \boldsymbol{u}^{(j)}\rangle] \\
&= \mathbb{E}[a_s\langle\nabla^{(j)} f(\boldsymbol{w}_{s,k,j}) - (\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1}) + \boldsymbol{\mu}_s^{(j)}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)}\rangle], \qquad (3.31)
\end{aligned}$$

where the first equality follows from $\boldsymbol{v}_{s,k-1}^{(j)} \in \mathcal{F}_{s,k,j-1}$ and the second equality follows from $\mathbb{E}[\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j})|\mathcal{F}_{s,k,j-1}] = \nabla^{(j)} f(\boldsymbol{w}_{s,k,j})$ and $\mathbb{E}[\nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1})|\mathcal{F}_{s,k,j-1}] = \nabla^{(j)} f(\tilde{\boldsymbol{y}}_{s-1}) = \boldsymbol{\mu}_s^{(j)}$. Meanwhile, for $j \in [m]$ and any fixed $\boldsymbol{u}^{(j)}$, we have

$$\begin{aligned}
&\mathbb{E}[a_{s,k-1}\langle\nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k-1,j}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{u}^{(j)}\rangle] \\
&= \mathbb{E}[a_{s,k-1}\langle\nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k-1,j}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)}\rangle] \\
&\quad + \mathbb{E}[\mathbb{E}[a_{s,k-1}\langle\nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k-1,j}), \boldsymbol{v}_{s,k-1}^{(j)} - \boldsymbol{u}^{(j)}\rangle|\mathcal{F}_{s,k,j-1}]] \\
&= \mathbb{E}[a_{s,k-1}\langle\nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k-1,j}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)}\rangle] \\
&\quad + \mathbb{E}[a_{s,k-1}\langle\nabla^{(j)} f(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)} f(\boldsymbol{w}_{s,k-1,j}), \boldsymbol{v}_{s,k-1}^{(j)} - \boldsymbol{u}^{(j)}\rangle], \qquad (3.32)
\end{aligned}$$

where the last equality is by $\boldsymbol{v}_{s,k-1}^{(j)} \in \mathcal{F}_{s,k,j-1}$, $\mathbb{E}[\nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k-1})|\mathcal{F}_{s,k,j-1}] = \nabla^{(j)} f(\boldsymbol{x}_{s,k-1})$ and $\mathbb{E}[\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k-1,j})|\mathcal{F}_{s,k,j-1}] = \nabla^{(j)} f(\boldsymbol{w}_{s,k-1,j})$. Combining Eqs. (3.30)–(3.32) completes the proof. □

In the following two lemmas, we will bound the third and the fourth terms of the R.H.S. in Lemma 20 by above using our novel Lipschitz Assumption 7 and Assumption 8.

Lemma 21. *For $s \geq 2$ and $k \in [K]$, we have*

$$\begin{aligned}
&-\sum_{j=1}^{m} a_{s,k-1}\mathbb{E}\left[\left\langle\nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k-1,j}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)}\right\rangle\right] \\
&\leq \mathbb{E}\left[\frac{K(1+A_{s-1}\gamma)}{8}\left\|\boldsymbol{v}_{s,k} - \boldsymbol{v}_{s,k-1}\right\|^2 + \frac{a_{s,k-1}^4 L^2}{KA_{s,k-1}^2(1+A_{s-1}\gamma)}\left\|\boldsymbol{v}_{s,k-1} - \boldsymbol{v}_{s,k-2}\right\|^2\right],
\end{aligned}$$

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

$$
\begin{aligned}
&- a_{s,k-1}\mathbb{E}\left[\left\langle \nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k-1,j}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)} \right\rangle\right] \\
&\leq \mathbb{E}\left[\frac{2a_{s,k-1}^2}{K\left(1+A_{s-1}\gamma\right)}\left\|\nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k-1,j})\right\|^2 + \frac{K\left(1+A_{s-1}\gamma\right)}{8}\left\|\boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)}\right\|^2\right] \\
&\leq \mathbb{E}\left[\frac{2a_{s,k-1}^2}{K\left(1+A_{s-1}\gamma\right)}\left\|\boldsymbol{x}_{s,k-1} - \boldsymbol{w}_{s,k-1,j}\right\|_{\boldsymbol{Q}^j}^2 + \frac{K\left(1+A_{s-1}\gamma\right)}{8}\left\|\boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)}\right\|^2\right] \\
&= \mathbb{E}\left[\frac{2a_{s,k-1}^2}{K\left(1+A_{s-1}\gamma\right)}\left\|\boldsymbol{x}_{s,k-1} - \boldsymbol{y}_{s,k-1}\right\|_{(\boldsymbol{Q}^j)_{\geq j+1}}^2 + \frac{K\left(1+A_{s-1}\gamma\right)}{8}\left\|\boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)}\right\|^2\right], \qquad (3.33)
\end{aligned}
$$

where we used Assumption 8 in the first inequality and the definitions of $\boldsymbol{x}_{s,k-1}$ and $\boldsymbol{w}_{s,k-1,j}$ in the last equality. Finally by including the summation and using the definition of L, $\boldsymbol{x}_{s,k-1}$ and $\boldsymbol{y}_{s,k-1}$, the first term of the above expression becomes

$$
\begin{aligned}
\sum_{j=1}^{m}\mathbb{E}\left[\frac{2a_{s,k-1}^2}{K\left(1+A_{s-1}\gamma\right)}\left\|\boldsymbol{x}_{s,k-1} - \boldsymbol{y}_{s,k-1}\right\|_{(\boldsymbol{Q}^j)_{\geq j+1}}^2\right] &= \mathbb{E}\left[\frac{2a_{s,k-1}^2}{K\left(1+A_{s-1}\gamma\right)}\left\|\boldsymbol{x}_{s,k-1} - \boldsymbol{y}_{s,k-1}\right\|_{\sum_{j=1}^{m}(\boldsymbol{Q}^j)_{\geq j+1}}^2\right] \\
&\leq \mathbb{E}\left[\frac{a_{s,k-1}^4 L^2}{KA_{s,k-1}^2\left(1+A_{s-1}\gamma\right)}\left\|\boldsymbol{v}_{s,k-1} - \boldsymbol{v}_{s,k-2}\right\|^2\right], \qquad (3.34)
\end{aligned}
$$

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. (3.33) and (3.34) give the lemma statement. □

Lemma 22. *For $s \geq 2$ and $k \in [K]$, we have*

$$
\begin{aligned}
&\sum_{j=1}^{m} a_s\mathbb{E}\left[\left\langle \nabla^{(j)} f(\boldsymbol{w}_{s,k,j}) - \left(\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1}) + \nabla^{(j)} f(\tilde{\boldsymbol{y}}_{s-1})\right), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)} \right\rangle\right] \\
&\leq \mathbb{E}\left[\left(\frac{2L^2 a_s^4}{KA_s^2(1+A_{s-1}\gamma)} + \frac{K(1+A_{s-1}\gamma)}{8}\right)\left\|\boldsymbol{v}_{s,k} - \boldsymbol{v}_{s,k-1}\right\|^2\right] \\
&\quad + \frac{8a_s^2 L}{K\left(1+A_{s-1}\gamma\right)}\mathbb{E}\left[f(\tilde{\boldsymbol{y}}_{s-1}) - f(\boldsymbol{x}_{s,k}) - \langle \nabla f(\boldsymbol{x}_{s,k}), \tilde{\boldsymbol{y}}_{s-1} - \boldsymbol{x}_{s,k}\rangle\right]
\end{aligned}
$$

Proof. Using similar arguments in the proof of Lemma 21, we have

$$
\begin{aligned}
&a_s\mathbb{E}\left[\left\langle \nabla^{(j)} f(\boldsymbol{w}_{s,k,j}) - \left(\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1}) + \nabla^{(j)} f(\tilde{\boldsymbol{y}}_{s-1})\right), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)} \right\rangle\right] \\
&\leq \mathbb{E}\left[\frac{2a_s^2}{K\left(1+A_{s-1}\gamma\right)}\left\|\nabla^{(j)} f(\boldsymbol{w}_{s,k,j}) - \left(\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{w,k,j}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1}) + \boldsymbol{\mu}_s^{(j)}\right)\right\|^2\right. \\
&\qquad\qquad \left. + \frac{K\left(1+A_{s-1}\gamma\right)}{8}\left\|\boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)}\right\|^2\right]
\end{aligned}
$$

$$
\begin{aligned}
&= \mathbb{E}\left[\frac{2a_s^2}{K\left(1+A_{s-1}\gamma\right)}\mathbb{E}\left[\left\|\left(\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) + \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1})\right) - \left(\nabla^{(j)} f(\boldsymbol{w}_{s,k,j}) - \boldsymbol{\mu}_s^{(j)}\right)\right\|^2 |\mathcal{F}_{s,k,j-1}\right]\right] \\
&\quad + \mathbb{E}\left[\frac{K\left(1+A_{s-1}\gamma\right)}{8}\left\|\boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)}\right\|^2\right] \\
&= \mathbb{E}\left[\frac{2a_s^2}{K\left(1+A_{s-1}\gamma\right)}\mathbb{E}\left[\left\|\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1})\right\|^2 |\mathcal{F}_{s,k,j-1}\right]\right] \\
&\quad + \mathbb{E}\left[\frac{K\left(1+A_{s-1}\gamma\right)}{8}\left\|\boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)}\right\|^2\right] \\
&\leq \mathbb{E}\left[\frac{4a_s^2}{K\left(1+A_{s-1}\gamma\right)}\left(\left\|\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k})\right\|^2 + \left\|\nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1})\right\|^2\right)\right. \\
&\qquad\qquad \left. + \frac{K\left(1+A_{s-1}\gamma\right)}{8}\left\|\boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{v}_{s,k-1}^{(j)}\right\|^2\right], \qquad (3.35)
\end{aligned}
$$

where the first equality comes from $\mathbb{E}\left[\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1})|\mathcal{F}_{s,k,j-1}\right] = \nabla^{(j)} f(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f(\tilde{\boldsymbol{y}}_{s-1})$ since the only randomness is in t_j when conditioned at $\mathcal{F}_{s,k,j-1}$, and the last inequality comes from $(a+b)^2 \leq 2(a^2+b^2)$. In order to bound the second term in Eq. (3.35), we will include the outer summation with respect to j and apply the results from Lemma 14 to get

$$
\begin{aligned}
\sum_{j=1}^{m}\mathbb{E}\left[\left\|\nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1})\right\|^2\right] &= \sum_{j=1}^{m}\mathbb{E}\left[\mathbb{E}\left[\left\|\nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k}) - \nabla^{(j)} f_{t_j}(\tilde{\boldsymbol{y}}_{s-1})\right\|^2 |\mathcal{F}_{s,k,0}\right]\right] \\
&= \mathbb{E}\left[\sum_{j=1}^{m}\sum_{l=1}^{n}\frac{1}{n}\left\|\nabla^{(j)} f_l(\boldsymbol{x}_{s,k}) - \nabla^{(j)} f_l(\tilde{\boldsymbol{y}}_{s-1})\right\|^2\right] \\
&= \mathbb{E}\left[\sum_{l=1}^{n}\frac{1}{n}\left\|\nabla f_l(\boldsymbol{x}_{s,k}) - \nabla f_l(\tilde{\boldsymbol{y}}_{s-1})\right\|^2\right] \\
&\leq \mathbb{E}\left[2L\left(f(\tilde{\boldsymbol{y}}_{s-1}) - f(\boldsymbol{x}_{s,k}) - \langle\nabla f(\boldsymbol{x}_{s,k}), \tilde{\boldsymbol{y}}_{s-1} - \boldsymbol{x}_{s,k}\rangle\right)\right], \qquad (3.36)
\end{aligned}
$$

where the second equality comes from $\boldsymbol{x}_{s,k}, \tilde{\boldsymbol{y}}_{s-1} \in \mathcal{F}_{s,k,0}$ and the last inequality is by applying Lemma 14 and the definition of $f(\boldsymbol{x}) = \frac{1}{n}\sum_{l=1}^{n} f_l(\boldsymbol{x})$. To bound the first term of Eq. (3.35), we apply similar arguments as in Lemma 21 and get

$$
\sum_{j=1}^{m}\mathbb{E}\left[\left\|\nabla^{(j)} f_{t_j}(\boldsymbol{w}_{s,k,j}) - \nabla^{(j)} f_{t_j}(\boldsymbol{x}_{s,k})\right\|^2\right] \leq \mathbb{E}\left[\frac{a_s^2 L^2}{2A_s^2}\left\|\boldsymbol{v}_{s,k} - \boldsymbol{v}_{s,k-1}\right\|^2\right]. \qquad (3.37)
$$

Combining Eqs. (3.35) – (3.37) gives the lemma statement. □

In the following lemma, we bound the expected error terms $\sum_{s=2}^{S}\sum_{k=1}^{K}\mathbb{E}\left[E_{s,k}(\boldsymbol{u})\right]$ arising from the gap bound stated in the previous lemma. This bound is then finally used in Theorem 7 to obtain the claimed convergence results.

Lemma 23. *With* $a_s^2 \leq \frac{KA_{s-1}(1+A_{s-1}\gamma)}{8L}$, $a_{s,k} = a_s$ *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* $\boldsymbol{u} \in \mathrm{dom}(g)$ *we have*

$$\begin{aligned}
&\sum_{s=2}^{S}\sum_{k=1}^{K}\mathbb{E}\left[E_{s,k}(\boldsymbol{u})\right] \\
&\leq -\sum_{j=1}^{m} a_1\left\langle \nabla^{(j)}f(\boldsymbol{x}_{1,1}) - \nabla^{(j)}f(\boldsymbol{w}_{1,1,j}), \boldsymbol{v}_{1,1}^{(j)} - \boldsymbol{u}^{(j)}\right\rangle \\
&\quad + \sum_{j=1}^{m} a_S\mathbb{E}\left[\left\langle \nabla^{(j)}f(\boldsymbol{x}_{S,K}) - \nabla^{(j)}f(\boldsymbol{w}_{S,K,j}), \boldsymbol{v}_{S,K}^{(j)} - \boldsymbol{u}^{(j)}\right\rangle\right] \\
&\quad + \frac{K}{64}\left\|\boldsymbol{v}_{1,1} - \boldsymbol{v}_{1,0}\right\|^2 - \frac{5K\left(1+A_{S-1}\gamma\right)}{32}\mathbb{E}\left[\left\|\boldsymbol{v}_{S,K} - \boldsymbol{v}_{S,K-1}\right\|^2\right],
\end{aligned}$$

where $\boldsymbol{x}_{1,1}, \boldsymbol{v}_{1,0} \in \mathrm{dom}(g)$ *can be chosen arbitrarily and* $\boldsymbol{w}_{1,1,j}$ *is defined in Algorithm 6.*

Proof. Combining Lemma 19, 20, 21, 22, setting a_s such that $a_s^2 = \frac{KA_{s-1}(1+A_{s-1}\gamma)}{8L}$ and using $\frac{A_{s-1}}{A_s} \leq 1$, we have

$$\begin{aligned}
\mathbb{E}\left[E_{s,k}(\boldsymbol{u})\right] \leq &\sum_{j=1}^{m} a_s\mathbb{E}\left[\left\langle \nabla^{(j)}f(\boldsymbol{x}_{s,k}) - \nabla^{(j)}f(\boldsymbol{w}_{s,k,j}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{u}^{(j)}\right\rangle\right] \\
&- \sum_{j=1}^{m} a_{s,k-1}\mathbb{E}\left[\left\langle \nabla^{(j)}f(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)}f(\boldsymbol{w}_{s,k-1,j}), \boldsymbol{v}_{s,k-1}^{(j)} - \boldsymbol{u}^{(j)}\right\rangle\right] \\
&- \left(\frac{5K\left(1+A_{s-1}\gamma\right)}{32}\right)\mathbb{E}\left[\left\|\boldsymbol{v}_{s,k} - \boldsymbol{v}_{s,k-1}\right\|^2\right] \\
&+ \left(\frac{a_{s,k-1}^4 L^2}{KA_{s,k-1}^2\left(1+A_{s-1}\gamma\right)}\right)\mathbb{E}\left[\left\|\boldsymbol{v}_{s,k-1} - \boldsymbol{v}_{s,k-2}\right\|^2\right].
\end{aligned}$$

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

$$\begin{aligned}
\sum_{s=2}^{S}\sum_{k=1}^{K}\mathbb{E}\left[E_{s,k}(\boldsymbol{u})\right] \leq & \sum_{j=1}^{m}\sum_{s=2}^{S}\sum_{k=1}^{K} a_s\mathbb{E}\left[\left\langle \nabla^{(j)}f(\boldsymbol{x}_{s,k}) - \nabla^{(j)}f(\boldsymbol{w}_{s,k,j}), \boldsymbol{v}_{s,k}^{(j)} - \boldsymbol{u}^{(j)}\right\rangle\right] \\
& - \sum_{j=1}^{m}\sum_{s=2}^{S}\sum_{k=1}^{K} a_{s,k-1}\mathbb{E}\left[\left\langle \nabla^{(j)}f(\boldsymbol{x}_{s,k-1}) - \nabla^{(j)}f(\boldsymbol{w}_{s,k-1,j}), \boldsymbol{v}_{s,k-1}^{(j)} - \boldsymbol{u}^{(j)}\right\rangle\right] \\
& - \sum_{s=2}^{S}\sum_{k=1}^{K}\frac{5K\left(1+A_{s-1}\gamma\right)}{32}\mathbb{E}\left[\left\|\boldsymbol{v}_{s,k} - \boldsymbol{v}_{s,k-1}\right\|^2\right] \\
& + \sum_{s=2}^{S}\left[\frac{K\left(1+A_{s-2}\gamma\right)}{64}\left\|\boldsymbol{v}_{s,0} - \boldsymbol{v}_{s,-1}\right\|^2 + \sum_{k=2}^{K}\frac{K\left(1+A_{s-1}\gamma\right)}{64}\left\|\boldsymbol{v}_{s,k-1} - \boldsymbol{v}_{s,k-2}\right\|^2\right] \\
\leq & \sum_{j=1}^{m} a_S\mathbb{E}\left[\left\langle \nabla^{(j)}f(\boldsymbol{x}_{S,K}) - \nabla^{(j)}f(\boldsymbol{w}_{S,K,j}), \boldsymbol{v}_{S,K}^{(j)} - \boldsymbol{u}^{(j)}\right\rangle\right] \\
& - \sum_{j=1}^{m} a_1\mathbb{E}\left[\left\langle \nabla^{(j)}f(\boldsymbol{x}_{1,1}) - \nabla^{(j)}f(\boldsymbol{w}_{1,1,j}), \boldsymbol{v}_{1,1}^{(j)} - \boldsymbol{u}^{(j)}\right\rangle\right] \\
& + \frac{K\left(1+A_0\gamma\right)}{64}\mathbb{E}\left[\left\|\boldsymbol{v}_{2,0} - \boldsymbol{v}_{2,-1}\right\|^2\right] - \frac{5K\left(1+A_{S-1}\gamma\right)}{32}\mathbb{E}\left[\left\|\boldsymbol{v}_{S,K} - \boldsymbol{v}_{S,K-1}\right\|^2\right].
\end{aligned}$$

The lemma follows by setting $A_0 = 0$, $\boldsymbol{v}_{2,-1} = \boldsymbol{v}_{1,0}$, $\boldsymbol{x}_{2,0} = \boldsymbol{x}_{1,1}$ and $\boldsymbol{w}_{2,0,j} = \boldsymbol{w}_{1,1,j}$. □

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

Theorem 7. *Let $\boldsymbol{x}_0 \in \mathrm{dom}(g)$ be an arbitrary initial point. Fix $K \geq 1$ and consider the updates in Algorithm 6. Then for $S \geq 2$ and $\forall \boldsymbol{u} \in \mathrm{dom}(g)$, we have*

$$\mathbb{E}\left[\bar{f}(\tilde{\boldsymbol{y}}_S) - \bar{f}(\boldsymbol{u})\right] + \frac{9\left(1+A_{S-1}\gamma\right)}{64A_S}\mathbb{E}\left[\left\|\boldsymbol{v}_{S,K} - \boldsymbol{u}\right\|^2\right] \leq \frac{5}{8A_S}\left\|\boldsymbol{x}_0 - \boldsymbol{u}\right\|^2.$$

In particular if $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \bar{f}(\boldsymbol{x})$ exists, then we have

$$\mathbb{E}\left[\bar{f}(\tilde{\boldsymbol{y}}_S) - \bar{f}(\boldsymbol{x}^*)\right] \leq \frac{5}{8A_S}\left\|\boldsymbol{x}_0 - \boldsymbol{x}^*\right\|^2$$

and

$$\mathbb{E}\left[\left\|\boldsymbol{v}_{S,K} - \boldsymbol{x}^*\right\|^2\right] \leq \frac{40}{9\left(1+A_{S-1}\gamma\right)}\left\|\boldsymbol{x}_0 - \boldsymbol{x}^*\right\|^2.$$

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 18 and Lemma 23, and by setting $\boldsymbol{y}_{1,0} = \boldsymbol{x}_{1,1} = \boldsymbol{x}_0$ and $\boldsymbol{y}_{1,1} = \boldsymbol{v}_{1,1}$, we have

$$\begin{aligned} KA_S\mathbb{E}\left[\bar{f}(\tilde{\boldsymbol{y}}_S) - \bar{f}(\boldsymbol{u})\right] \leq \frac{K}{2}\left\|\boldsymbol{x}_0 - \boldsymbol{u}\right\|^2 - \frac{K\left(1+A_S\right)}{2}\mathbb{E}\left[\left\|\boldsymbol{v}_{S,K} - \boldsymbol{u}\right\|^2\right] \\ - \frac{15K}{64}\left\|\boldsymbol{v}_{1,1} - \boldsymbol{x}_0\right\|^2 - \frac{5K\left(1+A_{S-1}\gamma\right)}{32}\mathbb{E}\left[\left\|\boldsymbol{v}_{S,K} - \boldsymbol{v}_{S,K-1}\right\|^2\right] \\ - \sum_{j=1}^{m} a_1\left\langle \nabla^{(j)} f(\boldsymbol{x}_{1,1}) - \nabla^{(j)} f(\boldsymbol{w}_{1,1,j}), \boldsymbol{v}_{1,1}^{(j)} - \boldsymbol{u}^{(j)}\right\rangle \\ + \sum_{j=1}^{m} a_S\mathbb{E}\left[\left\langle \nabla^{(j)} f(\boldsymbol{x}_{S,K}) - \nabla^{(j)} f(\boldsymbol{w}_{S,K,j}), \boldsymbol{v}_{S,K}^{(j)} - \boldsymbol{v}_{S,K-1}^{(j)}\right\rangle\right]. \end{aligned} \tag{3.38}$$

Using the same approach as Lemma 21 and Lemma 22, we can upper bound the first inner product term by

$$\begin{aligned} -\sum_{j=1}^{m} a_1\left\langle \nabla^{(j)} f(\boldsymbol{x}_{1,1}) - \nabla^{(j)} f(\boldsymbol{w}_{1,1,j}), \boldsymbol{v}_{1,1}^{(j)} - \boldsymbol{u}^{(j)}\right\rangle &\leq \frac{1}{8K}\left\|\boldsymbol{v}_{1,1} - \boldsymbol{x}_0\right\|^2 + \frac{K}{16}\left\|\boldsymbol{v}_{1,1} - \boldsymbol{u}\right\|^2 \\ &\leq \frac{15K}{64}\left\|\boldsymbol{v}_{1,1} - \boldsymbol{x}_0\right\|^2 + \frac{K}{8}\left\|\boldsymbol{x}_0 - \boldsymbol{u}\right\|^2, \end{aligned} \tag{3.39}$$

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

$$\begin{aligned} &\sum_{j=1}^{m} a_S\mathbb{E}\left[\left\langle \nabla^{(j)} f(\boldsymbol{x}_{S,K}) - \nabla^{(j)} f(\boldsymbol{w}_{S,K,j}), \boldsymbol{v}_{S,K}^{(j)} - \boldsymbol{v}_{S,K-1}^{(j)}\right\rangle\right] \\ &\quad\leq \frac{K\left(1+A_{S-1}\gamma\right)}{8}\mathbb{E}\left[\left\|\boldsymbol{v}_{S,K} - \boldsymbol{v}_{S,K-1}\right\|^2\right] + \frac{K\left(1+A_{S-1}\gamma\right)}{64}\mathbb{E}\left[\left\|\boldsymbol{v}_{S,K} - \boldsymbol{u}\right\|^2\right], \end{aligned} \tag{3.40}$$

where we also used $a_S^2 \leq \frac{KA_{S-1}(1+A_{S-1}\gamma)}{8L}$ here. Combining Eqs. (3.38)–(3.40) gives us our main bounds in the theorem. Lastly, recall that $\{a_s\}_{s\geq 1}$ is chosen so that ${a_s}^2 = \frac{KA_{s-1}(1+A_{s-1}\gamma)}{8L}$. When $\gamma = 0$, this leads to the standard $A_s \geq \frac{k^2K}{64L}$ growth of accelerated algorithms by choosing $a_s = \frac{sK}{32L}$ for $k \geq 1$. When $\gamma > 0$, we have $\frac{a_s}{A_{s-1}} > \sqrt{\frac{K\gamma}{8L}}$, and it remains to use that $A_k = \frac{A_k}{A_{k-1}} \cdot \ldots \cdot \frac{A_2}{A_1} \cdot A_1 = A_1\Big(1+\sqrt{\frac{K\gamma}{8L}}\Big)^{k-1}$ where $a_1 = A_1 = \frac{1}{4L}$ using the choice of a_k in Algorithm 3 and $A_0 = a_0 = 0$, completing the proof. □

Note that in Theorem 7, 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 $\tilde{\boldsymbol{y}}_S$ by applying Algorithm 5 such that $\mathbb{E}[\bar{f}(\tilde{\boldsymbol{y}}_S) - \bar{f}(\boldsymbol{x}^*)] \leq \epsilon$ is at most $O\Big(nd\sqrt{\frac{L\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}{n\epsilon}}\Big)$ for the general convex case when $\gamma = 0$, and $O\Big(\frac{nd\log(\epsilon L/\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|)}{\log(1+\sqrt{n\gamma/L})}\Big)$ for the strongly convex case when $\gamma > 0$.

3.4.1 Adaptive Variance Reduced A-CODER

Similar to A-CODER, VR-A-CODER can adaptively estimate the Lipschitz parameter by checking the quadratic bounds between $\boldsymbol{y}_s$, and $\boldsymbol{x}_{s,k}$ as well as between $\tilde{\boldsymbol{y}}_s$ and $\boldsymbol{x}_{s,k}$. For completeness, we have included the adaptive version of VR-A-CODER in Algorithm 7 below.

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 7.

3.5 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 3) and VR-A-CODER (Algorithm 5) 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 [SD21a], RCDM, ACDM [Nes12], ABCGD [BT13] and APCG [LLX15]. For all the methods, we use the function value gap $f(\boldsymbol{x}) - f(\boldsymbol{x}^*)$ 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. We set the block size to one in all the experiments, i.e., each block corresponds to one coordinate. We discussed in Section 3.4 that in theory it is optimal to choose $K = \Theta(n)$ 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 ($K \approx n/10$) 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 3.2), 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 3.3), 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 3.4), we use $\lambda_1 = 10^{-5}$ for sonar dataset and $\lambda_1 = 10^{-4}$ for a1a and a9a datasets. figures/acccyclic 3.3 and 3.4 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.

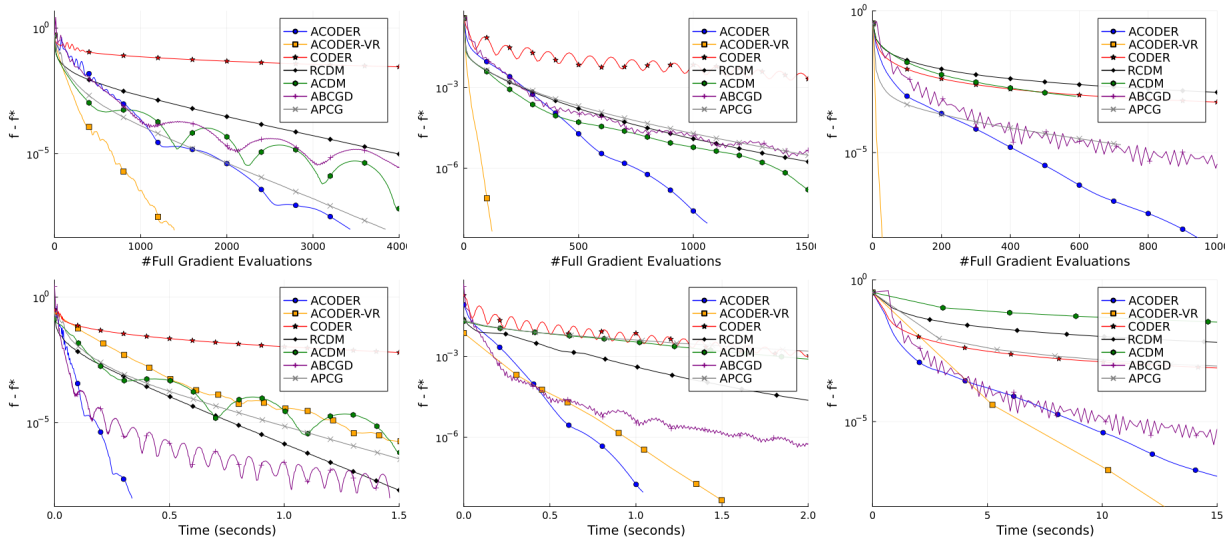


Figure 3.2: Performance comparisons between implemented algorithms in terms of the number of full-gradient evaluations and wall-clock time for logistic regression with ridge regularized problems. The top row contains plots against the number of full-gradient evaluations, and the bottom row contains plots 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 [CL11].

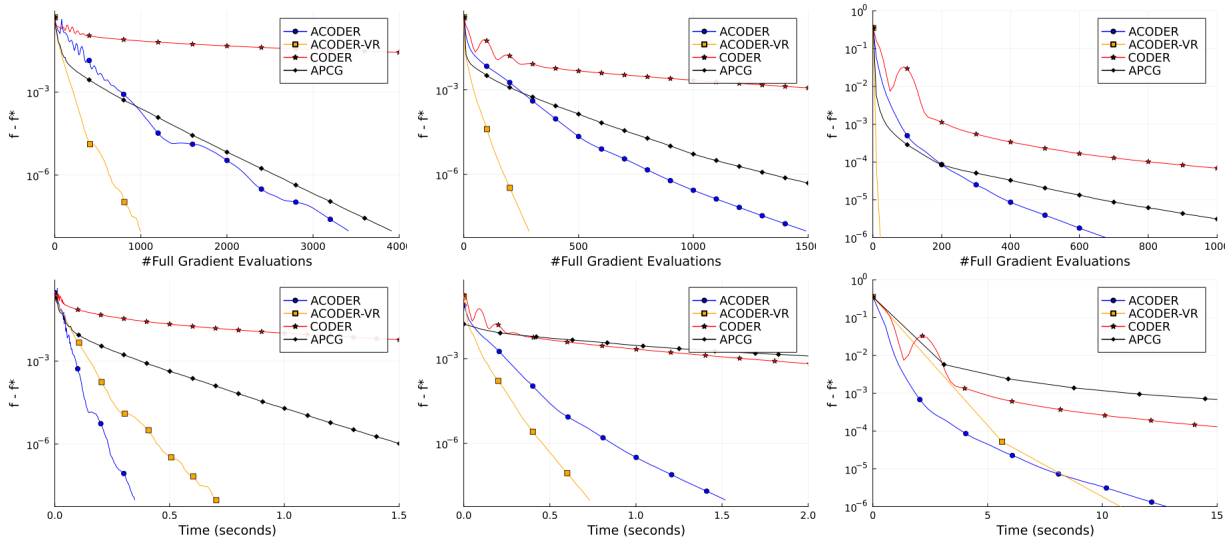


Figure 3.3: Performance comparisons between implemented algorithms in terms of the number of full-gradient evaluations and wall-clock time for logistic regression with elastic net regularized problems. The top row contains plots against the number of full-gradient evaluations, and the bottom row contains 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 [CL11].

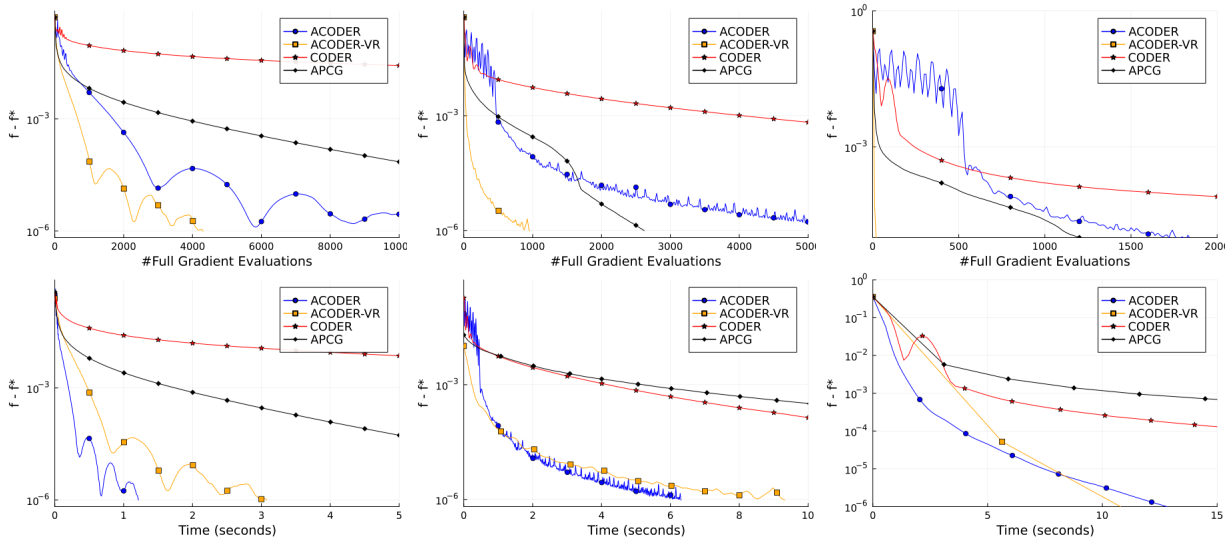


Figure 3.4: Performance comparisons between various algorithms in terms of number of full-gradient evaluations and wall-clock time for logistic regression with LASSO regularized problems. The top row contains plots against the number of full-gradient evaluations, and the bottom two contains plots against 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.

Algorithm 7 Variance Reduced A-CODER (Adaptive Version)

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

Chapter 4

Faster Algorithms for Solving Generalized Linear Programming and the Connection to Distributionally Robust Optimization

In this chapter, we study a class of generalized linear programs (GLP) in a large-scale setting, which includes a simple, possibly nonsmooth convex regularizer and simple convex set constraints. By reformulating GLP as an equivalent convex-concave min-max problem, we show that the linear structure in the problem can be used to design an efficient, scalable first-order algorithm, to which we give the name *Coordinate Linear Variance Reduction* (CLVR; pronounced "clever"). CLVR yields improved complexity results for GLP that depend on the max row norm of the linear constraint matrix in GLP rather than the spectral norm. When the regularization terms and constraints are separable, CLVR admits an efficient lazy update strategy that makes its complexity bounds scale with the number of nonzero elements of the linear constraint matrix in GLP rather than the matrix dimensions. Further, for the special case of linear programs and by exploiting sharpness, we propose a restart scheme for CLVR to obtain empirical linear convergence. Finally, we show that Distributionally Robust Optimization (DRO) problems with ambiguity sets based on both f-divergence and Wasserstein metrics can be reformulated as GLPs by introducing sparsely connected auxiliary variables. We complement our theoretical guarantees with numerical experiments that verify our algorithm's practical effectiveness in terms of wall-clock time and number of data passes.

4.1 Introduction

We study the following generalized linear program (GLP):

$$\min_{\boldsymbol{x}} \ \{\boldsymbol{c}^T\boldsymbol{x} + r(\boldsymbol{x}) : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \in \mathcal{X}\}, \tag{GLP}$$

where $\boldsymbol{x}, \boldsymbol{c} \in \mathbb{R}^d$, $\boldsymbol{A} \in \mathbb{R}^{n\times d}$, $\boldsymbol{b} \in \mathbb{R}^n$, $r : \mathbb{R}^d \to \mathbb{R}$ is a convex regularizer, and $\mathcal{X} \subseteq \mathbb{R}^d$ is a closed convex set, such that a proximal/projection operator involving r and $\mathcal{X}$ can be computed

efficiently. When $\mathcal{X}$ is the nonnegative orthant $\{\boldsymbol{x} : x_i \geq 0, i \in [d]\}$ and $r \equiv 0$, (GLP) reduces to the standard form of a linear program (LP). When $\mathcal{X}$ is a convex cone and $r \equiv 0$, (GLP) reduces to a conic linear program. (GLP) is an important paradigm in traditional engineering disciplines such as transportation, energy, telecommunications, and manufacturing. In modern data science, we note the renaissance of (GLP) due to its modeling power in such areas as reinforcement learning [DFVR03], optimal transport [Vil09], and neural network verification [LAL+20]. For traditional engineering disciplines with moderate scale or exploitable sparsity, off-the-shelf interior point methods that form and factorize matrices in each iteration are often good choices as practical solvers [Gur22]. In data science applications, however, where the data are often dense or of extreme scale, the amount of computation and/or memory required by matrix factorization is prohibitive. Thus, first-order methods that avoid matrix factorizations are potentially appealing options. In this context, because the presence of the linear equality constraint in (GLP) may complicate projection operations onto the feasible set, we consider an equivalent reformulation of (GLP) as a min-max problem involving the Lagrangian:

$$\min_{\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d} \max_{\boldsymbol{y} \in \mathbb{R}^n} \Big\{ \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) := \boldsymbol{c}^T \boldsymbol{x} + r(\boldsymbol{x}) + \boldsymbol{y}^T \boldsymbol{A} \boldsymbol{x} - \boldsymbol{y}^T \boldsymbol{b} \Big\}. \tag{PD-GLP}$$

In data science applications, both n and d can be very large. (PD-GLP) can be viewed as a structured bilinearly coupled min-max problem, where the linearity of $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$ in the dual variable vector $\boldsymbol{y}$ is vital to our algorithmic development.

4.1.1 Background

While there have been few papers that directly address (PD-GLP) — some special cases have been considered in [MM79, Man84, Man04, GZ17, Xu20, ZZ20, ZZ22, CJST20] — there has been significant recent work on first-order methods for general bilinearly coupled convex-concave min-max problems. Deterministic first-order methods include the proximal point method (PPM) [Roc76], the extragradient/mirror-prox method (EGM) [Kor76, Nem04], the primal-dual hybrid gradient (PDHG) method [CP11], and the alternating direction method of multipliers (ADMM) [DR56]. All these methods have per-iteration cost $\Theta(\text{nnz}(\boldsymbol{A}))$ and convergence rate $1/k$, where $\text{nnz}(\boldsymbol{A})$ denotes the number of nonzero elements of $\boldsymbol{A}$ and k is the number of iterations.

For better scalability, stochastic counterparts of these methods have been proposed. [JNT11, OHTG13, Bia16, PN17] have used "vanilla" stochastic gradients to replace the full gradients of their deterministic counterparts. [CJST19, HJ20, AM22] have exploited the finite-sum structure of the interaction term $\langle \boldsymbol{y}, \boldsymbol{A}\boldsymbol{x} \rangle$ involving both primal and dual variables to perform variance reduction. With a separability assumption for the dual variables, [ADFC17] and [CERS18] have combined incremental coordinate approaches on the dual variables with an implicit variance reduction strategy

on the primal variables. Recently, under a separability assumption for dual variables, [SWD21] proposed a new incremental coordinate method with an initialization step that requires a single access to the full data. This approach, known as *variance reduction via primal-dual accelerated dual averaging* (VRPDA^2), obtains the first theoretical bounds that are better than their deterministic counterparts in the class of incremental coordinate approaches. The VRPDA^2 algorithm serves as the main motivation for our approach.

It is of particular interest to design algorithms that scale with the number of nonzero elements in $\boldsymbol{A}$ for at least two reasons: (i) the data matrix can be sparse; and (ii) when we consider simplified reformulations of certain complicated models, we often need to introduce sparsely connected auxiliary variables. Nevertheless, the randomized coordinate algorithms of [ADFC17, CERS18, SWD21] have $O(d)$ per-iteration cost regardless of the sparsity of $\boldsymbol{A}$. To address this issue, [FB19, LFP19] have proposed incremental primal-dual coordinate methods with per-iteration cost that scales with the number of nonzero elements in the row of $\boldsymbol{A}$ used in each iteration, at the price of needing to take a smaller step than for dense $\boldsymbol{A}$. Moreover, [AFC20] has proposed a random extrapolation approach that admits both low per-iteration cost and larger step size. Despite these developments, all these algorithms produce less accurate iterates than the methods with $O(d)$ per-iteration cost, thus degrading their worst-case complexity. [1]

Finally, for the special case of LP, based on the positive Hoffman constant [Hof03], [AHLL21] proved that the primal-dual formulation of LP exhibits a sharpness property that lower-bounds the growth of a normalized primal-dual gap from the same work. Leveraging this sharpness property, [AHLL21] proposed a restart scheme for the deterministic first-order methods discussed above to obtain linear convergence. [ADH$^+$21] further extended this restart strategy using various heuristics to improve practical performance.

4.1.2 Motivation

We sharpen the focus from general bilinearly coupled convex-concave min-max problems to (GLP) and its primal-dual formulation (PD-GLP), because many complicated models can be reformulated as (GLP) and because this formulation possesses additional structure that can be exploited in algorithm design. Our motivation for focusing on (GLP) is to bridge the large gap between the well-studied stochastic variance reduced first-order methods [JZ13, AZ17, SJM20, SWD21] and the increasingly popular and complicated, yet highly structured large-scale problems arising in distributionally robust optimization (DRO) [WKS14, SAMEK15, ND16, EK18, HNSS18, DN21, LHS19, DGN21, YLMJ21]; see also a recent survey by [RM19] and references therein.

[1] Subsequent to this paper, a version of the PURE-CD algorithm of [AFC20] that exploits sparsity in $\boldsymbol{A}$ was developed and analyzed in [ACW22].

For DRO problems with ambiguity sets defined by f-divergence [ND16, HNSS18, LCDS20], the original formulation is a nonbilinearly coupled convex-concave min-max problem. Even the well constructed reformulation in [LCDS20] does not admit unbiased stochastic gradients, leading to complicated algorithms and analysis. For DRO problems with ambiguity sets defined by Wasserstein metric [SAMEK15, EK18, LHS19, YLMJ21, HNW22], the original formulation is in general infinite-dimensional. (Finite-dimensional reformulations [SAMEK15, EK18] exist for special cases of logistic regression and smooth convex losses.) Solvers that have been proposed for DRO with Wasserstein metric are either multiple-loop deterministic ADMM [LHS19] or are designed for general convex-concave problems [YLMJ21].

By introducing auxiliary variables with sparse connections, [2] we show that DRO with ambiguity sets based on both f-divergence and the Wasserstein metric can be reformulated as (GLP). Thus, complicated DRO problems can be addressed by a simple, efficient, and scalable algorithm for (GLP). Our algorithm for solving (GLP) and the proposed reformulations of DRO are our main contributions.

4.2 Contributions

Algorithm. Motivated by VRPDA^2 [SWD21], we propose a simple, efficient, and scalable algorithm for (PD-GLP). Our algorithm combines an incremental *coordinate* method with exploitation of the *linear* structure for the dual variables in (PD-GLP) and the implicit *variance reduction* effect in the algorithm, so we name it *coordinate linear variance reduction* (CLVR, pronounced "clever"). CLVR is inspired by VRPDA^2 but customized to the particular structure of (PD-GLP). In particular, by exploiting the fact that the max problem is linear and unconstrained in the dual variable vector $\boldsymbol{y} \in \mathbb{R}^n$, we find that the expensive initialization step used in VRPDA^2 is not needed and we can take simpler and larger steps. Further, in the structured case in which $\boldsymbol{A}$ is sparse and the convex constraint set $\mathcal{X}$ and the regularizer $r(\boldsymbol{x})$ are fully separable[3], we show that the dual averaging update in CLVR enables us to design an efficient lazy update strategy for which the per-iteration cost of CLVR scales with the number of nonzero elements of the selected row from $\boldsymbol{A}$ in each iteration, which is potentially much lower than the order-d cost in VRPDA^2. Finally, CLVR uses extrapolation on dual variables rather than on primal variables considered in VRPDA^2, which significantly reduces implementation complexity of our lazy update strategy for structured variants of (PD-GLP). On the technical side, although both CLVR and VRPDA^2 are randomized algorithms that bound the

[2] "Sparse connections" here means that even though the newly introduced variables may substantially increase the problem dimensions, the number of nonzero entries in the constraint matrix remains of the same order.

[3] We state the results here for the fully separable setting for convenience of comparison; however, our results are also applicable to the block separable setting.

primal-dual gap in expectation, the guarantee provided by CLVR is stronger as it allows bounding the expectation of the supremum gap as opposed to the supremum of expected gap in VRPDA^2.

To state our complexity results, we make the following scaling assumption.

Assumption 9. *$L := \|\boldsymbol{A}\|$ and each row of $\boldsymbol{A}$ in* (GLP) *is normalized with Euclidean norm R.*

Preprocessing in modern LP solvers [Gur22] often ensures normalized rows/columns for the data matrix. Observe that $R \leq L \leq \sqrt{n}R$, the upper bound being achieved when all elements of $\boldsymbol{A}$ have identical value. Although the latter case is extreme, there exist ill-conditioned practical datasets where we can expect significant performance gains if the complexity can be reduced from $O(L)$ to $O(R)$. (We provide empirical comparison between the values of L and R in practical problems in Section 4.6.)

In Table 4.1, we give the overall complexity bounds (total number of arithmetic operations) and the per-iteration cost of a representative set of existing algorithms, including our CLVR algorithm, for solving a structured form of (PD-GLP) in which the set $\mathcal{X}$ and the function r have separable structure: $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$ with $\mathcal{X}_i \in \mathbb{R}\,(i \in [d])$ and $r(\boldsymbol{x}) := \sum_{i=1}^{d} r(x^i)$. To make the complexity results comparable, we assume further that for the stochastic algorithms [CERS18, AM22, SWD21] and our CLVR algorithm, we draw one row of $\boldsymbol{A}$ per iteration uniformly at random. The general convex setting corresponds to $r(\boldsymbol{x})$ being general convex ($\sigma = 0$), while the strongly convex setting corresponds to $r(\boldsymbol{x})$ being σ-strongly convex ($\sigma > 0$).

As shown in Table 4.1, all the algorithms have optimal dependence on ϵ [OX19], while the dependence on the ambient dimensions n, d, the number of nonzero elements of $\boldsymbol{A}$ (nnz($\boldsymbol{A}$)), and the constants L and R are quite different. For both the general convex and strongly convex settings and among coordinate-type methods, CLVR is the first algorithm that reduces the runtime dependence on the input matrix size from nd to nnz($\boldsymbol{A}$). Moreover, the complexity of CLVR depends on the max row norm R rather than the spectral norm L, and the per-iteration cost of CLVR depends only on the nonzero elements of the selected row from $\boldsymbol{A}$ in each iteration, which can be far less than d.

By exploiting the linear structure again, we provide explicit guarantees for both the objective value and the constraint satisfaction of (GLP). Further, the analysis of CLVR applies to the more general *block-coordinate* update setting, which is better suited to modern parallel computing platforms. Finally, following the restart strategy based on the *normalized duality gap* for LP introduced in [AHLL21], we propose a more straightforward strategy to restart our CLVR algorithm (as well as other iterative algorithms for (PD-GLP)): Restart the algorithm every time a widely known metric for LP optimality [AA00] halves. Compared with the normalized duality gap, the LPMetric can be computed more efficiently and in a more straightforward fashion.

Table 4.1: Overall complexity and per-iteration cost for solving structured (PD-GLP). ("—" indicates that the corresponding result does not exist or is unknown.)

Algorithm	General Convex (Primal-Dual Gap)	Strongly Convex (Distance to Solution)	Per-Iteration Cost
PDHG [CP11]	$O(\frac{\text{nnz}(\boldsymbol{A})L}{\epsilon})$	$O(\frac{(\text{nnz}(\boldsymbol{A})+n+d)L}{\sigma\sqrt{\epsilon}})$	$O(\text{nnz}(\boldsymbol{A}))$
SPDHG [CERS18]	$O(\frac{ndL}{\epsilon})$	$O(\frac{ndL}{\sigma\sqrt{\epsilon}})$	$O(d)$
EVR [AM22]	$O(\text{nnz}(\boldsymbol{A}) + \frac{\sqrt{\text{nnz}(\boldsymbol{A})(n+d)nR}}{\epsilon})$	—	$O(n+d)$
VRPDA^2 SWD([SWD21])	$O(nd\log\min\{\frac{1}{\epsilon}, n\} + \frac{ndR}{\epsilon})$	$O(nd\log\min\{\frac{1}{\epsilon}, n\} + \frac{ndR}{\sigma\sqrt{\epsilon}})$	$O(d)$
CLVR (**This Paper**)	$O(\frac{\text{nnz}(\boldsymbol{A})R}{\epsilon})$	$O(\frac{\text{nnz}(\boldsymbol{A})R}{\sigma\sqrt{\epsilon}})$	$O(\text{nnz}(\text{row}(\boldsymbol{A})))$

DRO reformulations. When the loss function is convex, DRO problems with ambiguity sets based on f-divergence [ND16] or Wasserstein metric [EK18] are convex. However, because both problems either have complicated constraints or are infinite-dimensional, vanilla first-order methods are inapplicable.

For DRO with f-divergence, we show that by using convex conjugates and introducing auxiliary variables, the problem can be reformulated as a (GLP). As a result, the issue of biased stochastic gradients encountered in [LCDS20] does not arise, and CLVR can be applied. Even though the resulting problem has larger dimensions, due to the sparseness of the introduced auxiliary variables and the lazy update strategy of CLVR, it can be solved with complexity scaling only with the number of nonzero elements of the data matrix. Due to being cast as a (GLP), the DRO problem can be solved with $O(1/\epsilon)$ iteration complexity with CLVR, while existing methods such as [LCDS20] have $O(1/\epsilon^2)$ iteration complexity, with higher iteration cost because of the batch of samples needed to reduce bias. This improvement is enabled in part by considering the primal-dual gap (rather than the primal gap considered in [LCDS20]) and by allowing the constraints to be approximately satisfied (see Corollary 1).

For DRO with Wasserstein metric, following the reformulation of [SAMEK15, Theorem 1], we show further that the problem can be cast in the form of (GLP). Compared with the existing reformulations [SAMEK15, EK18, LHS19, YLMJ21], our reformulation can handle both smooth and nonsmooth convex loss functions. In fact, our reformulation can provide a more compact form for nonsmooth piecewise-linear convex loss functions (such as hinge loss). Moreover, compared with

algorithms customized to this problem [LHS19] and extragradient methods [Kor76, Nem04, YLMJ21] for general convex-concave min-max problems, our CLVR method attains the best-known iteration complexity and per-iteration cost, as shown in Table 4.1.

4.3 Notation and preliminaries

For any positive integer p, we use $[p]$ to denote $\{1, 2, \ldots, p\}$. We assume that there is a given partition of the set $[n]$ into sets S^j, $j \in [m]$, where $|S^j| = n^j > 0$ and $\sum_{j=1}^m n^j = n$. For $j \in [m]$, we use $\boldsymbol{A}^{S^j}$ to denote the submatrix of $\boldsymbol{A}$ with rows indexed by S^j and $\boldsymbol{y}^{S^j}$ to denote the subvector of $\boldsymbol{y}$ indexed by S^j. We use $\mathbf{0}_d$ and $\mathbf{1}_d$ to denote the vectors with all ones and all zeros in d dimensions, respectively. Unless otherwise specified, we use $\|\cdot\|$ to denote the Euclidean norm for vectors and the spectral norm for matrices. For a given proper convex lower semi-continuous function $f : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$, we define the convex conjugate in the standard way as $f^*(y) = \sup_{x \in \mathbb{R}} \{yx - f(x)\}$ (so that $f^{**} = f$). For a vector $\boldsymbol{u}$, the inequality $\boldsymbol{u} \geq \mathbf{0}$ is applied entry-wise. For a convex function $r(\boldsymbol{x})$, we use $r'(\boldsymbol{x})$ to denote an element of the subdifferential set $\partial r(\boldsymbol{x})$. The proximal operator of $r(\boldsymbol{x})$ over $\mathcal{X}$ is

$$\text{prox}_r(\hat{\boldsymbol{x}}) = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \Big\{ \frac{1}{2} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 + r(\boldsymbol{x}) \Big\}. \tag{4.1}$$

Further, we make the following assumptions, which apply throughout the convergence analysis.

Assumption 10. (PD-GLP) *attains at least one primal-dual solution* $(\boldsymbol{x}^*, \boldsymbol{y}^*)$. $\mathcal{W}^*$ *denotes the set of all primal-dual solutions.*

Due to the convex-concave property of (PD-GLP), $\mathcal{W}^*$ is a convex set in $\mathcal{X} \times \mathbb{R}^n$.

Assumption 11. $\hat{L} = \max_{j \in [m]} \|\boldsymbol{A}^{S^j}\|$ *is given at the input, where* $\|\boldsymbol{A}^{S^j}\| = \max_{\|\boldsymbol{x}\| \leq 1} \|\boldsymbol{A}^{S^j} \boldsymbol{x}\|$.

Note that $\hat{L}$ can be obtained either via preprocessing of the data or by parameter tuning. By combining Assumptions 9 and 11, it follows that $R \leq \hat{L} \leq \sqrt{\max_{j \in [m]} |S^j|} R$.

Assumption 12. $r(\boldsymbol{x})$ *is* σ*-strongly convex* ($\sigma \geq 0$)*; that is, for all* $\boldsymbol{x}_1$ *and* $\boldsymbol{x}_2$ *in* $\mathcal{X}$ *and all* $r'(\boldsymbol{x}_2) \in \partial r(\boldsymbol{x}_2)$*, we have* $r(\boldsymbol{x}_1) \geq r(\boldsymbol{x}_2) + \langle r'(\boldsymbol{x}_2), \boldsymbol{x}_1 - \boldsymbol{x}_2 \rangle + \frac{\sigma}{2} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2$.

For convex-concave min-max problems, a common metric for measuring solution quality is the primal-dual gap, which, for a feasible solution $(\boldsymbol{x}, \boldsymbol{y})$ of (PD-GLP), is defined by

$$\sup_{(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{X} \times \mathbb{R}^n} \{\mathcal{L}(\boldsymbol{x}, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \boldsymbol{y})\}. \tag{4.2}$$

However, as the domain of $\boldsymbol{v}$ is unbounded, the primal-dual gap can be infinite, which makes it a poor metric for measuring the progress of algorithms. As a result, for measuring the progress of our algorithm, we consider the following restricted primal-dual gap instead:

$$\sup_{(\boldsymbol{u},\boldsymbol{v})\in\mathcal{W}} \{\mathcal{L}(\boldsymbol{x},\boldsymbol{v}) - \mathcal{L}(\boldsymbol{u},\boldsymbol{y})\}, \tag{4.3}$$

where $\mathcal{W} \subset \mathcal{X} \times \mathbb{R}^n$ is a compact (i.e., closed and bounded) convex set. The use of a restricted version of primal-dual gap is standard in the existing literature; see, e.g., [Nes07a, CP11].

4.4 The CLVR algorithm

We specify the implementation of CLVR in Algorithm 8 for (PD-GLP) in the general setting. However, the implementation version can be difficult to understand from a theoretical perspective directly due to the closed form format for $\boldsymbol{y}_k^{S^i}$ where it stems from an minimization problem of the estimation sequence $\psi(\boldsymbol{y})$. As such, we state a version of CLVR in Algorithm 9 that is convenient for analysis, and is equivalent to Algorithm 8. Before analyzing the convergence of CLVR, we justify our claim of equivalence in Proposition 1.

Algorithm 8 Coordinate Linear Variance Reduction (CLVR)

Input: $\boldsymbol{x}_0 \in \mathcal{X}, \boldsymbol{y}_0 \in \mathbb{R}^n, \boldsymbol{z}_0 = \boldsymbol{A}^T\boldsymbol{y}_0, \gamma > 0, \hat{L} > 0, \sigma \geq 0, K, m, \{S^1, S^2, \ldots, S^m\}$.
$a_1 = A_1 = \frac{1}{2\hat{L}m}, \boldsymbol{q}_0 = a_1(\boldsymbol{z}_0 + \boldsymbol{c})$.
for $k = 1, 2, \ldots, K$ **do**
  $\boldsymbol{x}_k = \mathrm{prox}_{\frac{1}{\gamma}A_k r}(\boldsymbol{x}_0 - \frac{1}{\gamma}\boldsymbol{q}_{k-1})$.
  Pick j_k uniformly at random in $[m]$.
  $\boldsymbol{y}_k^{S^i} = \begin{cases} \boldsymbol{y}_{k-1}^{S^i}, & i \neq j_k \\ \boldsymbol{y}_{k-1}^{S^i} + \gamma m a_k(\boldsymbol{A}^{S^i}\boldsymbol{x}_k - \boldsymbol{b}^{S^i}), & i = j_k \end{cases}$.
  $a_{k+1} = \frac{\sqrt{1+\sigma A_k/\gamma}}{2\hat{L}m}, A_{k+1} = A_k + a_{k+1}$.
  $\boldsymbol{z}_k = \boldsymbol{z}_{k-1} + \boldsymbol{A}^{S^{j_k},T}(\boldsymbol{y}_k^{S^{j_k}} - \boldsymbol{y}_{k-1}^{S^{j_k}})$.
  $\boldsymbol{q}_k = \boldsymbol{q}_{k-1} + a_{k+1}(\boldsymbol{z}_k + \boldsymbol{c}) + m a_k(\boldsymbol{z}_k - \boldsymbol{z}_{k-1})$.
end for
return $\tilde{\boldsymbol{x}}_K = \frac{1}{A_K}\sum_{k=1}^K a_k\boldsymbol{x}_k$, $\tilde{\boldsymbol{y}}_K = \frac{1}{A_K}\sum_{k=1}^K (a_k\boldsymbol{y}_k + (m-1)a_k(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}))$.

Proposition 1. *The iterates of Algorithm 8 and 9 are equivalent.*

Proof. To argue equivalence, we show that the iterates of Algorithm 8 and 9 solve the same optimization problems. To avoid ambiguity, here we will use $\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{y}}_k$ to denote the iterates $\boldsymbol{x}_k, \boldsymbol{y}_k$ in Algorithm 9, while we retain the notation $\boldsymbol{x}_k, \boldsymbol{y}_k$ for the iterates of Algorithm 8.

Let us first start by writing an equivalent definition of $\boldsymbol{x}_k$ in Algorithm 8. To do so, we first unroll the recursive definitions of $\boldsymbol{z}_k$ and $\boldsymbol{q}_k$. We can observe that, since by definition $\boldsymbol{y}_k$ and $\boldsymbol{y}_{k-1}$ only differ over the coordinate block S^{j_k}, we have

$$\boldsymbol{z}_k = \boldsymbol{z}_0 + \sum_{i=1}^{k} \boldsymbol{A}^T(\boldsymbol{y}_i - \boldsymbol{y}_{i-1}) = \boldsymbol{A}^T\boldsymbol{y}_k. \tag{4.4}$$

On the other hand, using Eq. (4.4), the definition of $\boldsymbol{q}_k$ implies

$$\boldsymbol{q}_k = A_{k+1}\boldsymbol{c} + a_1\boldsymbol{A}^T\boldsymbol{y}_0 + \sum_{i=1}^{k} \boldsymbol{A}^T[a_{i+1}\boldsymbol{y}_i + ma_i(\boldsymbol{y}_i - \boldsymbol{y}_{i-1})] \tag{4.5}$$

Using the definition of the proximal operator (see Eq. (4.1)) and the definition of $\boldsymbol{x}_k$ in Step 4 of Algorithm 8, we have

$$\begin{aligned} \boldsymbol{x}_k &= \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \frac{A_k}{\gamma} r(\boldsymbol{x}) + \frac{1}{2}\Big\|\boldsymbol{x} - \boldsymbol{x}_0 + \frac{1}{\gamma}\boldsymbol{q}_{k-1}\Big\|^2 \right\} \\ &= \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ A_k r(\boldsymbol{x}) + \frac{\gamma}{2}\Big\|\boldsymbol{x} - \boldsymbol{x}_0 + \frac{1}{\gamma}\boldsymbol{q}_{k-1}\Big\|^2 \right\}. \end{aligned} \tag{4.6}$$

Now let us consider the optimization problem that defines $\hat{\boldsymbol{x}}_k$. Assume for now that the definitions of $\boldsymbol{y}_k$ and $\hat{\boldsymbol{y}}_k$ agree (we justify this claim below). Observe first that the minimization problem defining $\hat{\boldsymbol{x}}_k$ is independent of $\boldsymbol{u}$, so by unrolling the recursion for ϕ_k, we have

$$\begin{aligned} \hat{\boldsymbol{x}}_k &= \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \frac{\gamma}{2}\|\boldsymbol{x} - \boldsymbol{x}_0\|^2 + A_k r(\boldsymbol{x}) + a_1 \left\langle \boldsymbol{x}, \boldsymbol{c} + \boldsymbol{A}^T\bar{\boldsymbol{y}}_0 \right\rangle \right. \\ &\qquad \left. + \sum_{i=2}^{k} a_i \left\langle \boldsymbol{x}, \boldsymbol{c} + \boldsymbol{A}^T\Big(\boldsymbol{y}_{i-1} + \frac{ma_{i-1}}{a_i}(\boldsymbol{y}_{i-1} - \boldsymbol{y}_{i-2})\Big) \right\rangle \right\} \\ &= \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \frac{\gamma}{2}\|\boldsymbol{x} - \boldsymbol{x}_0\|^2 + A_k r(\boldsymbol{x}) \right. \\ &\qquad \left. + \left\langle \boldsymbol{x}, A_k\boldsymbol{c} + a_1\boldsymbol{A}^T\boldsymbol{y}_0 + \boldsymbol{A}^T\Big(\sum_{i=2}^{k}[a_i\boldsymbol{y}_{i-1} + ma_{i-1}(\boldsymbol{y}_{i-1} - \boldsymbol{y}_{i-2})]\Big) \right\rangle \right\} \\ &= \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \frac{\gamma}{2}\|\boldsymbol{x} - \boldsymbol{x}_0\|^2 + A_k r(\boldsymbol{x}) + \langle \boldsymbol{x}, \boldsymbol{q}_{k-1}\rangle \right\} \\ &= \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ A_k r(\boldsymbol{x}) + \frac{\gamma}{2}\Big\|\boldsymbol{x} - \boldsymbol{x}_0 + \frac{1}{\gamma}\boldsymbol{q}_{k-1}\Big\|^2 \right\} \\ &= \boldsymbol{x}_k. \end{aligned}$$

It remains to argue that the definitions of $\boldsymbol{y}_k$ and $\hat{\boldsymbol{y}}_k$ agree. First, observe that since the definitions of ψ_k and ψ_{k-1} differ only over block S^{j_k}, we have $\hat{\boldsymbol{y}}_k^{S^j} = \hat{\boldsymbol{y}}_{k-1}^{S^j}$ for all $j \neq j_k$. For $j = j_k$, we have by

unrolling the recursive definition of ψ_k that

$$\begin{aligned}\hat{\boldsymbol{y}}_k^{S^{j_k}} &= \underset{\boldsymbol{y}^{S^{j_k}} \in \mathbb{R}^{n_{j_k}}}{\arg\min} \left\{ \frac{1}{2\gamma} \|\boldsymbol{y}^{S^{j_k}} - \boldsymbol{y}_0^{S^{j_k}}\|^2 - \sum_{i=1}^{k} \mathbb{1}_{\{S^{j_i}=S^{j_k}\}} m a_i \left\langle \boldsymbol{y}^{S^{j_i}}, \boldsymbol{A}^{S^{j_i}} \boldsymbol{x}_i - \boldsymbol{b}^{S^{j_i}} \right\rangle \right\} \\ &= \boldsymbol{y}_0^{S^{j_k}} + \gamma \sum_{i=1}^{k} \mathbb{1}_{\{S^{j_i}=S^{j_k}\}} m a_i (\boldsymbol{A}^{S^{j_i}} \boldsymbol{x}_i - \boldsymbol{b}^{S^{j_i}}) \\ &= \hat{\boldsymbol{y}}_{k-1}^{S^{j_k}} + \gamma m a_k (\boldsymbol{A}^{S^{j_k}} \boldsymbol{x}_k - \boldsymbol{b}^{S^{j_k}}) \\ &= \boldsymbol{y}_k^{S^{j_k}},\end{aligned}$$

as claimed. □

Algorithm 9 Coordinate Linear Variance Reduction (Analysis Version)

1: **Input:** $\boldsymbol{x}_0 = \boldsymbol{x}_{-1} \in \mathcal{X}, \boldsymbol{y}_0 = \bar{\boldsymbol{y}}_0 \in \mathbb{R}^n$, $m, \{S^1, S^2, \ldots, S^m\}, K, \gamma > 0, \hat{L} > 0$.
2: $\phi_0(\cdot) = \frac{\gamma}{2}\|\cdot - \boldsymbol{x}_0\|^2, \psi_0(\cdot) = \frac{1}{2\gamma}\|\cdot - \boldsymbol{y}_0\|^2$.
3: $a_1 = A_1 = \frac{1}{2\hat{L}m}$.
4: **for** $k = 1, 2, 3, \ldots, K$ **do**
5: $\quad \boldsymbol{x}_k = \arg\min_{\boldsymbol{x}\in\mathcal{X}}\{\phi_k(\boldsymbol{x}) = \phi_{k-1}(\boldsymbol{x}) + a_k(\langle \boldsymbol{x} - \boldsymbol{u}, \boldsymbol{A}^T \bar{\boldsymbol{y}}_{k-1} + \boldsymbol{c}\rangle + r(\boldsymbol{x}) - r(\boldsymbol{u}))\}$.
6: $\quad$ Pick j_k uniformly at random in $[m]$.
7: $\quad \boldsymbol{y}_k = \arg\min_{\boldsymbol{y}\in\mathbb{R}^n}\{\psi_k(\boldsymbol{y}) = \psi_{k-1}(\boldsymbol{y}) + m a_k(-\langle \boldsymbol{y}^{S^{j_k}} - \boldsymbol{v}^{S^{j_k}}, \boldsymbol{A}^{S^{j_k}} \boldsymbol{x}_k - \boldsymbol{b}^{S^{j_k}}\rangle)\}$.
8: $\quad a_{k+1} = \frac{\sqrt{1+\sigma A_k/\gamma}}{2\hat{L}m}, A_{k+1} = A_k + a_{k+1}$.
9: $\quad \bar{\boldsymbol{y}}_k = \boldsymbol{y}_k + \frac{m a_k}{a_{k+1}}(\boldsymbol{y}_k - \boldsymbol{y}_{k-1})$.
10: **end for**
11: **return** $\tilde{\boldsymbol{x}}_K := \frac{1}{A_K}\sum_{k=1}^{K} a_k \boldsymbol{x}_k, \tilde{\boldsymbol{y}}_K = \frac{1}{A_K}\sum_{k=1}^{K}(a_k \boldsymbol{y}_k + (m-1)a_k(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}))$.

4.4.1 Algorithm and analysis for general formulation

The algorithm alternates the full update for $\boldsymbol{x}_k$ in Step 4 ($O(d)$ cost) with an incremental block coordinate update for $\boldsymbol{y}_k$ in Steps 5 and 6 (with $O(|S^{j_k}|d)$ cost for dense $\boldsymbol{A}$). The auxiliary variables $\boldsymbol{z}_k$ and $\boldsymbol{q}_k$ accumulate the cancellation terms in the estimation sequence and give a pathway to a straightforward development of the lazified CLVR, which appears as Algorithm 10 in Section 4.4.2. The cost of updating auxiliary vectors $\boldsymbol{z}_k$ and $\boldsymbol{q}_k$ is $O(|S^{j_k}|d)$ and $O(d)$, respectively. In essence, CLVR is a primal-dual coordinate method that uses a *dual averaging* update for $\boldsymbol{x}_k$, then updates the state variables $\{\boldsymbol{q}_k\}$ by a *linear recursion*, and computes $\boldsymbol{x}_k$ from $\boldsymbol{q}_{k-1}$ via a *proximal step* without direct dependence on $\boldsymbol{x}_{k-1}$. The output $\tilde{\boldsymbol{x}}_K$ is a convex combination of the iterates $\{\boldsymbol{x}_k\}_{k=1}^K$, as is standard for primal-dual methods. However, $\tilde{\boldsymbol{y}}_K$ is only an *affine* (not convex) combination

of $\{\boldsymbol{y}_k\}_{k=0}^K$, as it involves the term $-(m-1)\boldsymbol{y}_0$ (whose coefficient is negative) and some of the coefficients $ma_k - (m-1)a_{k+1}$ multiplying $\boldsymbol{y}_k$ for $k \in \{1, \dots K-1\}$ may also be negative. An affine combination still provides valid bounds because the dual variable vector $\boldsymbol{y}$ appears linearly in (PD-GLP). Moreover, in Step 9, the term $ma_k(\boldsymbol{z}_k - \boldsymbol{z}_{k-1})$ serves to cancel certain errors from the randomization of the update w.r.t. $\boldsymbol{y}_k$, thus playing a key role in implicit variance reduction.

In the following three lemmas, we bound $\phi_k(\boldsymbol{x}_k)$ and $\psi_k(\boldsymbol{y}_k)$ below and above, which is then subsequently used to bound the primal-dual gap in Theorem 8.

Lemma 24. *For all steps of Algorithm 9 with $k \geq 1$, we have, $\forall(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{X} \times \mathbb{R}^n$,*

$$\begin{aligned} \phi_k(\boldsymbol{x}_k) &\leq \frac{\gamma}{2}\|\boldsymbol{u} - \boldsymbol{x}_0\|^2 - \frac{\gamma + \sigma A_k}{2}\|\boldsymbol{u} - \boldsymbol{x}_k\|^2, \\ \psi_k(\boldsymbol{y}_k) &\leq \frac{1}{2\gamma}\|\boldsymbol{v} - \boldsymbol{y}_0\|^2 - \frac{1}{2\gamma}\|\boldsymbol{v} - \boldsymbol{y}_k\|^2. \end{aligned}$$

Proof. By the definitions of $\psi_k(\boldsymbol{y})$ and $\phi_k(\boldsymbol{x})$ in Algorithm 9, it follows that, $\forall k \geq 1$,

$$\phi_k(\boldsymbol{x}) = \sum_{i=1}^{k} a_i(\langle \boldsymbol{x} - \boldsymbol{u}, \boldsymbol{A}^T\bar{\boldsymbol{y}}_{i-1} + \boldsymbol{c}\rangle + r(\boldsymbol{x}) - r(\boldsymbol{u})) + \frac{\gamma}{2}\|\boldsymbol{x} - \boldsymbol{x}_0\|^2, \tag{4.7}$$

and

$$\psi_k(\boldsymbol{y}) = \sum_{i=1}^{k} ma_i\Big(- \Big\langle \boldsymbol{y}^{S^{j_i}} - \boldsymbol{v}^{S^{j_i}}, \boldsymbol{A}^{S^{j_i}}\boldsymbol{x}_i - \boldsymbol{b}^{S^{j_i}} \Big\rangle \Big) + \frac{1}{2\gamma}\|\boldsymbol{y} - \boldsymbol{y}_0\|^2. \tag{4.8}$$

Observe that, as function of $\boldsymbol{x}$, $\phi_k(\boldsymbol{x})$ is $(\gamma + \sigma A_k)$-strongly convex. As, by definition, $\boldsymbol{x}_k = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \phi_k(\boldsymbol{x})$, it follows that

$$\phi_k(\boldsymbol{u}) \geq \phi_k(\boldsymbol{x}_k) + \frac{\gamma + \sigma A_k}{2}\|\boldsymbol{u} - \boldsymbol{x}_k\|^2. \tag{4.9}$$

Now, writing $\phi_k(\boldsymbol{u})$ explicitly and rearranging the last inequality, the stated bound on $\phi_k(\boldsymbol{x}_k)$ follows.

As a function of $\boldsymbol{y}$, $\psi_k(\boldsymbol{y})$ is $1/\gamma$-strongly convex. The proof for the bound on ψ_k uses the same argument and is omitted. □

In the following proof, for $k \geq 1$, let $\mathcal{F}_k$ denote the natural filtration, containing all the randomness in the algorithm up to and including iteration k. Recall that $\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}^{S^1} \\ \vdots \\ \boldsymbol{A}^{S^m} \end{pmatrix}$, and let $\boldsymbol{A}^{\bar{S}^j}(j \in [m])$ denote the matrix $\boldsymbol{A}$ with its S^j block of rows replaced by a zero block.

For convenience, for $k = 1, 2, \dots$, we define

$$\hat{\boldsymbol{y}}_k = \boldsymbol{y}_{k-1} + \gamma ma_k(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}). \tag{4.10}$$

Then from the definition of $\boldsymbol{y}_k$ in Algorithm 8, we have $\mathbb{E}[\boldsymbol{y}_k - \boldsymbol{y}_{k-1}|\mathcal{F}_{k-1}] = \frac{1}{m}(\hat{\boldsymbol{y}}_k - \boldsymbol{y}_{k-1})$.

Motivated by [AFC19], we have the following result.

Lemma 25. *Given the sequences $\{\boldsymbol{y}_k\}$ in Algorithm 9 and $\{\hat{\boldsymbol{y}}_k\}$ in Eq. (4.10), we define the sequence $\{\check{\boldsymbol{v}}_k\}$ by*

$$\check{\boldsymbol{v}}_k = (\boldsymbol{y}_k - \boldsymbol{y}_{k-1}) - \frac{1}{m}(\hat{\boldsymbol{y}}_k - \boldsymbol{y}_{k-1}).$$

Then for any $\boldsymbol{v} \in \mathbb{R}^d$ that may be correlated with the randomness in $\{\check{\boldsymbol{v}}_i\}_{i=1}^k$, we have

$$\mathbb{E}\Big[-\sum_{i=1}^k \langle \boldsymbol{v}, \check{\boldsymbol{v}}_i\rangle\Big] \le \mathbb{E}\Big[\frac{1}{2}\|\boldsymbol{y}_0 - \boldsymbol{v}\|^2 + \frac{1}{2}\sum_{i=1}^k \|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2\Big], \tag{4.11}$$

where the expectation is taken over all the randomness in the history.

Proof. First, we prove a bound for $\mathbb{E}\Big[\sum_{i=1}^k \|\check{\boldsymbol{v}}_i\|^2\Big]$. By the fact that $\mathbb{E}[\boldsymbol{y}_i - \boldsymbol{y}_{i-1}|\mathcal{F}_{i-1}] = \frac{1}{m}(\hat{\boldsymbol{y}}_i - \boldsymbol{y}_{i-1})$, for each $\check{\boldsymbol{v}}_i$, we have

$$\mathbb{E}[\|\check{\boldsymbol{v}}_i\|^2|\mathcal{F}_{i-1}] = \mathbb{E}\Big[\|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2|\mathcal{F}_{i-1}\Big] - \mathbb{E}\Big[\|\frac{1}{m}(\hat{\boldsymbol{y}}_i - \boldsymbol{y}_{i-1})\|^2|\mathcal{F}_{i-1}\Big] \le \mathbb{E}[\|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2|\mathcal{F}_{i-1}].$$

Taking expectation on all the randomness in the history, for $\mathbb{E}\Big[\sum_{i=1}^k \|\check{\boldsymbol{v}}_i\|^2\Big]$, we have

$$\mathbb{E}\Big[\sum_{i=1}^k \|\check{\boldsymbol{v}}_i\|^2\Big] = \mathbb{E}\Big[\sum_{i=1}^k \mathbb{E}[\|\check{\boldsymbol{v}}_i\|^2|\mathcal{F}_{i-1}]\Big] \le \mathbb{E}\Big[\sum_{i=1}^k \mathbb{E}[\|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2|\mathcal{F}_{i-1}]\Big] = \mathbb{E}\Big[\sum_{i=1}^k \|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2\Big]. \tag{4.12}$$

Then to prove (4.11), we define the sequence $\{\check{\boldsymbol{y}}_k\}_{k=0}^\infty$ by $\check{\boldsymbol{y}}_0 = \boldsymbol{y}_0$ and $\check{\boldsymbol{y}}_k = \check{\boldsymbol{y}}_{k-1} - \check{\boldsymbol{v}}_k$ for $k = 1, 2, \ldots$. By expanding $\frac{1}{2}\|\check{\boldsymbol{y}}_k - \boldsymbol{v}\|^2$,

$$\begin{aligned} \frac{1}{2}\|\check{\boldsymbol{y}}_k - \boldsymbol{v}\|^2 &= \frac{1}{2}\|\check{\boldsymbol{y}}_{k-1} - \boldsymbol{v}\|^2 - \langle \check{\boldsymbol{v}}_k, \check{\boldsymbol{y}}_{k-1} - \boldsymbol{v}\rangle + \frac{1}{2}\|\check{\boldsymbol{v}}_k\|^2 \\ \implies \quad \langle \check{\boldsymbol{v}}_k, \check{\boldsymbol{y}}_{k-1} - \boldsymbol{v}\rangle &= \frac{1}{2}\|\check{\boldsymbol{y}}_{k-1} - \boldsymbol{v}\|^2 - \frac{1}{2}\|\check{\boldsymbol{y}}_k - \boldsymbol{v}\|^2 + \frac{1}{2}\|\check{\boldsymbol{v}}_k\|^2. \end{aligned}$$

By summing this expression and telescoping, we obtain

$$\begin{aligned} \sum_{i=1}^k \langle \check{\boldsymbol{y}}_{i-1} - \boldsymbol{v}, \check{\boldsymbol{v}}_i\rangle &= \frac{1}{2}\|\check{\boldsymbol{y}}_0 - \boldsymbol{v}\|^2 - \frac{1}{2}\|\check{\boldsymbol{y}}_k - \boldsymbol{v}\|^2 + \sum_{i=1}^k \frac{1}{2}\|\check{\boldsymbol{v}}_i\|^2 \\ &\le \frac{1}{2}\|\check{\boldsymbol{y}}_0 - \boldsymbol{v}\|^2 + \sum_{i=1}^k \frac{1}{2}\|\check{\boldsymbol{v}}_i\|^2 \\ &= \frac{1}{2}\|\boldsymbol{y}_0 - \boldsymbol{v}\|^2 + \sum_{i=1}^k \frac{1}{2}\|\check{\boldsymbol{v}}_i\|^2, \end{aligned} \tag{4.13}$$

It follows that

$$\begin{aligned} \mathbb{E}\Big[-\sum_{i=1}^k \langle \boldsymbol{v}, \check{\boldsymbol{v}}_i\rangle\Big] &= \mathbb{E}\Big[\sum_{i=1}^k \langle \check{\boldsymbol{y}}_{i-1} - \boldsymbol{v}, \check{\boldsymbol{v}}_i\rangle - \sum_{i=1}^k \langle \check{\boldsymbol{y}}_{i-1}, \check{\boldsymbol{v}}_i\rangle\Big] \\ &\le \mathbb{E}\Big[\frac{1}{2}\|\boldsymbol{y}_0 - \boldsymbol{v}\|^2 + \sum_{i=1}^k \frac{1}{2}\|\check{\boldsymbol{v}}_i\|^2 - \sum_{i=1}^k \langle \check{\boldsymbol{y}}_{i-1}, \check{\boldsymbol{v}}_i\rangle\Big] \\ &\le \mathbb{E}\Big[\frac{1}{2}\|\boldsymbol{y}_0 - \boldsymbol{v}\|^2 + \frac{1}{2}\sum_{i=1}^k \|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2 - \sum_{i=1}^k \langle \check{\boldsymbol{y}}_{i-1}, \check{\boldsymbol{v}}_i\rangle\Big]. \end{aligned} \tag{4.14}$$

where the first inequality is by Eq. (4.13) and the second inequality is by Eq. (4.12). To obtain the result (4.11), we use the facts that $\mathbb{E}[\check{\boldsymbol{v}}_i \,|\, \mathcal{F}_{i-1}] = 0$ and that $\check{\boldsymbol{y}}_{i-1} \in \mathcal{F}_{i-1}$ to obtain $\mathbb{E}\Big[\sum_{i=1}^k \langle \check{\boldsymbol{y}}_{i-1}, \check{\boldsymbol{v}}_i \rangle\Big] = 0$. □

We emphasize that Lemma 25 holds for any $\boldsymbol{v}$ that may be even correlated with the randomness in the algorithm.

Lemma 26. *For all steps of Algorithm 9 with $k \geq 1$, we have for all $(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{X} \times \mathbb{R}^n$ that*

$$\begin{aligned}
\phi_k(\boldsymbol{x}_k) \geq \phi_{k-1}(\boldsymbol{x}_{k-1}) + \frac{\gamma + \sigma A_{k-1}}{2}\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|^2 + a_k(r(\boldsymbol{x}_k) - r(\boldsymbol{u})) \\
- a_k\Big\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_k - \boldsymbol{y}_{k-1})\Big\rangle + a_k\Big\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T\boldsymbol{y}_k + \boldsymbol{c}\Big\rangle \\
+ ma_{k-1}\Big(\Big\langle \boldsymbol{x}_{k-1} - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2})\Big\rangle + \Big\langle \boldsymbol{x}_k - \boldsymbol{x}_{k-1}, \boldsymbol{A}^T(\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2})\Big\rangle\Big), \\
\psi_k(\boldsymbol{y}_k) \geq \psi_{k-1}(\boldsymbol{y}_{k-1}) + \frac{1}{2\gamma}\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2 - a_k\langle \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}, \boldsymbol{y}_k - \boldsymbol{v}\rangle \\
- (m-1)a_k\langle \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}, \boldsymbol{y}_k - \boldsymbol{y}_{k-1}\rangle + \frac{1}{\gamma}\Big\langle \boldsymbol{y}_{k-1} - \boldsymbol{v}, -(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}) + \frac{1}{m}(\hat{\boldsymbol{y}}_k - \boldsymbol{y}_{k-1})\Big\rangle.
\end{aligned}$$

Proof. For the first claim, we have from the definition of $\phi_k(\boldsymbol{x}_k)$, using that $\phi_{k-1}(\boldsymbol{x}_{k-1})$ is $(\gamma+\sigma A_{k-1})$-strongly convex and minimized at $\boldsymbol{x}_{k-1}$, that

$$\begin{aligned}
\phi_k(\boldsymbol{x}_k) \geq \phi_{k-1}(\boldsymbol{x}_{k-1}) + \frac{\gamma + \sigma A_{k-1}}{2}\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|^2 \\
+ a_k(\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T\bar{\boldsymbol{y}}_{k-1} + \boldsymbol{c}\rangle + r(\boldsymbol{x}_k) - r(\boldsymbol{u})).
\end{aligned} \tag{4.15}$$

Meanwhile, by the definition of $\{\bar{\boldsymbol{y}}_k\}$ (using $\boldsymbol{y}_{-1} = \boldsymbol{y}_0$ for the case of $k = 1$), we have that

$$\begin{aligned}
& a_k\Big\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T\bar{\boldsymbol{y}}_{k-1}\Big\rangle \\
&= a_k\Big\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T\Big(\boldsymbol{y}_{k-1} + \frac{ma_{k-1}}{a_k}(\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2})\Big)\Big\rangle \\
&= -a_k\Big\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_k - \boldsymbol{y}_{k-1})\Big\rangle + a_k\Big\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T\boldsymbol{y}_k\Big\rangle \\
&\quad + ma_{k-1}\Big(\Big\langle \boldsymbol{x}_{k-1} - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2})\Big\rangle + \Big\langle \boldsymbol{x}_k - \boldsymbol{x}_{k-1}, \boldsymbol{A}^T(\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2})\Big\rangle\Big).
\end{aligned} \tag{4.16}$$

The claimed lower bound on $\phi_k(\boldsymbol{x}_k)$ follows when we combine (4.15) and (4.16).

For the second claim, we have by the definition of ψ_k that

$$\psi_k(\boldsymbol{y}_k) - \psi_{k-1}(\boldsymbol{y}_{k-1}) = \psi_{k-1}(\boldsymbol{y}_k) - \psi_{k-1}(\boldsymbol{y}_{k-1}) - ma_k\Big\langle \boldsymbol{y}_k^{S^{j_k}} - \boldsymbol{v}^{S^{j_k}}, \boldsymbol{A}^{S^{j_k}}\boldsymbol{x}_k - \boldsymbol{b}^{S^{j_k}}\Big\rangle. \tag{4.17}$$

To obtain the claimed lower bound on ψ_k, we proceed to bound the terms on the right-hand side in (4.17). First, since ψ_{k-1} is $(1/\gamma)$-strongly convex and minimized at $\boldsymbol{y}_{k-1}$, we have

$$\psi_{k-1}(\boldsymbol{y}_k) - \psi_{k-1}(\boldsymbol{y}_{k-1}) \geq \frac{1}{2\gamma}\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2. \tag{4.18}$$

Second, by using the definition of $\boldsymbol{A}^{S_{j_k}}$ and $\boldsymbol{A}^{\bar{S}_{j_k}}$, and by using several times that $\boldsymbol{y}_{k-1}$ and $\boldsymbol{y}_k$ differ only in their S^{j_k} components, we have

$$
\begin{aligned}
&-\left\langle \boldsymbol{y}_k^{S^{j_k}} - \boldsymbol{v}^{S^{j_k}}, \boldsymbol{A}^{S^{j_k}}\boldsymbol{x}_k - \boldsymbol{b}^{S^{j_k}} \right\rangle \\
&\quad = -\left\langle \boldsymbol{y}_k - \boldsymbol{v}, \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} \right\rangle + \left\langle \boldsymbol{y}_k - \boldsymbol{v}, \boldsymbol{A}^{\bar{S}^{j_k}}\boldsymbol{x}_k - \boldsymbol{b}^{\bar{S}^{j_k}} \right\rangle \\
&\quad = -\left\langle \boldsymbol{y}_k - \boldsymbol{v}, \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} \right\rangle + \left\langle \boldsymbol{y}_{k-1} - \boldsymbol{v}, \boldsymbol{A}^{\bar{S}^{j_k}}\boldsymbol{x}_k - \boldsymbol{b}^{\bar{S}^{j_k}} \right\rangle \\
&\quad = -\left\langle \boldsymbol{y}_k - \boldsymbol{v}, \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} \right\rangle + \frac{m-1}{m}\left\langle \boldsymbol{y}_{k-1} - \boldsymbol{v}, \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} \right\rangle \\
&\qquad + \left\langle \boldsymbol{y}_{k-1} - \boldsymbol{v}, \boldsymbol{A}^{\bar{S}^{j_k}}\boldsymbol{x}_k - \boldsymbol{b}^{\bar{S}^{j_k}} - \frac{m-1}{m}(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}) \right\rangle \\
&\quad = -\frac{1}{m}\left\langle \boldsymbol{y}_k - \boldsymbol{v}, \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} \right\rangle + \frac{m-1}{m}\left\langle \boldsymbol{y}_{k-1} - \boldsymbol{y}_k, \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} \right\rangle \\
&\qquad + \left\langle \boldsymbol{y}_{k-1}^{S^{j_k}} - \boldsymbol{v}^{S^{j_k}}, -(\boldsymbol{A}^{S^{j_k}}\boldsymbol{x}_k - \boldsymbol{b}^{S^{j_k}}) \right\rangle + \frac{1}{m}\left\langle \boldsymbol{y}_{k-1} - \boldsymbol{v}, \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} \right\rangle \rangle \\
&\quad = -\frac{1}{m}\left\langle \boldsymbol{y}_k - \boldsymbol{v}, \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} \right\rangle + \frac{m-1}{m}\left\langle \boldsymbol{y}_{k-1} - \boldsymbol{y}_k, \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} \right\rangle \\
&\qquad + \left\langle \boldsymbol{y}_{k-1} - \boldsymbol{v}, -\frac{1}{\gamma m a_k}(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}) + \frac{1}{\gamma m^2 a_k}(\hat{\boldsymbol{y}}_k - \boldsymbol{y}_{k-1}) \right\rangle,
\end{aligned}
\tag{4.19}
$$

where in the last equality we used $\boldsymbol{y}_k^{S^{j_k}} - \boldsymbol{y}_{k-1}^{S^{j_k}} = \gamma m a_k(\boldsymbol{A}^{S^{j_k}}\boldsymbol{x}_k - \boldsymbol{b}^{S^{j_k}})$ and $\hat{\boldsymbol{y}}_k - \boldsymbol{y}_{k-1} = \gamma m a_k(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b})$, which both hold by definitions of $\hat{\boldsymbol{y}}_k$ and $\boldsymbol{y}_k$.

Finally, combining (4.17)–(4.19), we have the bound on $\psi_k(\boldsymbol{y}_k)$ from the statement of the lemma.

□

By combining the two lower bounds in Lemma 26, we obtain the following result.

Lemma 27. *For any $(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{X} \times \mathbb{R}^n$, the sum of $\psi_k(\boldsymbol{y}_k) + \phi_k(\boldsymbol{x}_k)$ can be bounded as follows: for all $k \geq 1$,*

$$
\begin{aligned}
&\phi_k(\boldsymbol{x}_k) + \psi_k(\boldsymbol{y}_k) \\
&\geq \phi_{k-1}(\boldsymbol{x}_{k-1}) + \psi_{k-1}(\boldsymbol{y}_{k-1}) \\
&\quad - m a_k \left\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}) \right\rangle + m a_{k-1} \left\langle \boldsymbol{x}_{k-1} - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2}) \right\rangle \\
&\quad + \frac{1}{2\gamma}\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2 - \frac{1}{4\gamma}\|\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2}\|^2 + Q_k,
\end{aligned}
\tag{4.20}
$$

where

$$
\begin{aligned}
Q_k := a_k \Big(& r(\boldsymbol{x}_k) - r(\boldsymbol{u}) - \langle \boldsymbol{A}\boldsymbol{u}, \boldsymbol{y}_k \rangle + \langle \boldsymbol{x}_k, \boldsymbol{A}^T\boldsymbol{v} \rangle + \langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{c} \rangle + \langle \boldsymbol{b}, \boldsymbol{y}_k - \boldsymbol{v} \rangle \\
& - (m-1)\langle \boldsymbol{A}\boldsymbol{u} - \boldsymbol{b}, \boldsymbol{y}_k - \boldsymbol{y}_{k-1} \rangle \Big) + \frac{1}{\gamma}\left\langle \boldsymbol{y}_{k-1} - \boldsymbol{v}, -(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}) + \frac{1}{m}(\hat{\boldsymbol{y}}_k - \boldsymbol{y}_{k-1}) \right\rangle.
\end{aligned}
\tag{4.21}
$$

Proof. Before our proof, as A_{-1} and A_0 are not used in Algorithm 9, without loss of generality, we set $A_{-1} = A_0 = 0$. Fix any $(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{X} \times \mathbb{R}^n$. By combining the bounds on $\phi_k(\boldsymbol{x}_k)$ and $\psi_k(\boldsymbol{y}_k)$ from Lemma 26, we have $\forall k \geq 1$ that

$$\begin{aligned}
&\phi_k(\boldsymbol{x}_k) + \psi_k(\boldsymbol{y}_k) \\
\geq\ &\phi_{k-1}(\boldsymbol{x}_{k-1}) + \psi_{k-1}(\boldsymbol{y}_{k-1}) \\
&- ma_k \left\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}) \right\rangle + ma_{k-1} \left\langle \boldsymbol{x}_{k-1} - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2}) \right\rangle \\
&+ P_k + Q_k,
\end{aligned} \tag{4.22}$$

where

$$P_k = \frac{\gamma + \sigma A_{k-1}}{2} \|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|^2 + \frac{1}{2\gamma} \|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2 + ma_{k-1} \left\langle \boldsymbol{x}_k - \boldsymbol{x}_{k-1}, \boldsymbol{A}^T(\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2}) \right\rangle \tag{4.23}$$

and Q_k is defined in Eq. (4.21).

To find a lower bound on P_k we start by bounding the magnitude of the inner product term in (4.23). Recall that $\boldsymbol{y}_{k-2}$ and $\boldsymbol{y}_{k-1}$ differ only on the coordinate block $S^{j_{k-1}}$. We thus have:

$$\begin{aligned}
&|ma_{k-1} \langle \boldsymbol{x}_k - \boldsymbol{x}_{k-1}, \boldsymbol{A}^T(\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2}) \rangle| \\
&= |ma_{k-1} \ \langle \boldsymbol{A}^{S^{j_{k-1}}}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}), \boldsymbol{y}_{k-1}^{S^{j_{k-1}}} - \boldsymbol{y}_{k-2}^{S^{j_{k-1}}} \rangle| \\
&\leq ma_{k-1} \|\boldsymbol{A}^{S^{j_{k-1}}}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})\| \|\boldsymbol{y}_{k-1}^{S^{j_{k-1}}} - \boldsymbol{y}_{k-2}^{S^{j_{k-1}}}\| \\
&\leq (ma_{k-1})^2 \gamma \|\boldsymbol{A}^{S^{j_{k-1}}}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})\|^2 + \frac{1}{4\gamma} \|\boldsymbol{y}_{k-1}^{S^{j_{k-1}}} - \boldsymbol{y}_{k-2}^{S^{j_{k-1}}}\|^2 \\
&\leq (m\hat{L}a_{k-1})^2 \gamma \|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|^2 + \frac{1}{4\gamma} \|\boldsymbol{y}_{k-1}^{S^{j_{k-1}}} - \boldsymbol{y}_{k-2}^{S^{j_{k-1}}}\|^2 \\
&= (m\hat{L}a_{k-1})^2 \gamma \|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|^2 + \frac{1}{4\gamma} \|\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2}\|^2
\end{aligned} \tag{4.24}$$

where the second inequality is by Young's inequality, and the third inequality is by Assumption 11 where $\|\boldsymbol{A}^{S^{j_{k-1}}}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})\| \leq \hat{L}\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|$. With our setting $a_{k-1} = \frac{\sqrt{1+\sigma A_{k-2}/\gamma}}{2m\hat{L}} \leq \frac{\sqrt{1+\sigma A_{k-1}/\gamma}}{2m\hat{L}}$, for all $k \geq 1$, we have

$$(m\hat{L}a_{k-1})^2 \gamma = \frac{\gamma + \sigma A_{k-1}}{4}.$$

By substituting this equality into (4.24) and then combining with (4.23), we obtain

$$\begin{aligned}
P_k &\geq \frac{\gamma + \sigma A_{k-1}}{4} \|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|^2 + \frac{1}{2\gamma} \|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2 - \frac{1}{4\gamma} \|\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2}\|^2 \\
&\geq \frac{1}{2\gamma} \|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2 - \frac{1}{4\gamma} \|\boldsymbol{y}_{k-1} - \boldsymbol{y}_{k-2}\|^2.
\end{aligned} \tag{4.25}$$

We complete the proof by combining Eqs. (4.22), (4.23) and the bound Eq. (4.25) for P_k. □

By telescoping the inequality in Lemma 27, and using Lemmas 24 and 25, we obtain the next result.

Lemma 28. *For all* $(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{X} \times \mathbb{R}^n$, *we have*

$$\begin{aligned} & A_k(\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \tilde{\boldsymbol{y}}_k)) + \frac{\gamma + \sigma A_k}{4}\|\boldsymbol{u} - \boldsymbol{x}_k\|^2 + \frac{1}{2\gamma}\|\boldsymbol{v} - \boldsymbol{y}_k\|^2 \\ \leq\ & \frac{\gamma}{2}\|\boldsymbol{u} - \boldsymbol{x}_0\|^2 + \frac{1}{2\gamma}\|\boldsymbol{v} - \boldsymbol{y}_0\|^2 - \frac{1}{4\gamma}\sum_{i=1}^{k}\|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2 + \frac{1}{\gamma}\sum_{i=1}^{k}\langle \boldsymbol{y}_{i-1}, \check{\boldsymbol{v}}_i\rangle - \frac{1}{\gamma}\sum_{i=1}^{k}\langle \boldsymbol{v}, \check{\boldsymbol{v}}_i\rangle, \end{aligned} \tag{4.26}$$

where $\check{\boldsymbol{v}}_i$ *is defined in Lemma 25.*

Proof. Telescoping the inequality in Lemma 27, we have

$$\begin{aligned} \phi_k(\boldsymbol{x}_k) + \psi_k(\boldsymbol{y}_k) \geq\ & \phi_0(\boldsymbol{x}_0) + \psi_0(\boldsymbol{y}_0) \\ & - ma_k\left\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_k - \boldsymbol{y}_{k-1})\right\rangle + ma_0\left\langle \boldsymbol{x}_0 - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_0 - \boldsymbol{y}_{-1})\right\rangle \\ & + \frac{1}{2\gamma}\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2 - \frac{1}{4\gamma}\|\boldsymbol{y}_0 - \boldsymbol{y}_{-1}\|^2 + \frac{1}{4\gamma}\sum_{i=1}^{k-1}\|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2 + \sum_{i=1}^{k} Q_i \\ =\ & - ma_k\left\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_k - \boldsymbol{y}_{k-1})\right\rangle + \frac{1}{4\gamma}\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2 \\ & + \frac{1}{4\gamma}\sum_{i=1}^{k}\|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2 + \sum_{i=1}^{k} Q_i, \end{aligned} \tag{4.27}$$

where the last equality is by the fact that $\psi_0(\boldsymbol{y}_0) = \phi_0(\boldsymbol{x}_0) = 0$, $a_0 = 0$, and our convention that $\boldsymbol{y}_{-1} := \boldsymbol{y}_0$.

Then by the convexity of $r(\cdot)$ and the definition of $\{Q_i\}$, we have

$$\begin{aligned} \sum_{i=1}^{k} Q_i \geq\ & A_k(r(\tilde{\boldsymbol{x}}_k) - r(\boldsymbol{u}) - \langle \boldsymbol{A}\boldsymbol{u}, \tilde{\boldsymbol{y}}_k\rangle + \langle \tilde{\boldsymbol{x}}_k, \boldsymbol{A}^T\boldsymbol{v}\rangle + \langle \tilde{\boldsymbol{x}}_k - \boldsymbol{u}, \boldsymbol{c}\rangle + \langle \boldsymbol{b}, \tilde{\boldsymbol{y}}_k - \boldsymbol{v}\rangle) \\ & + \frac{1}{\gamma}\sum_{i=1}^{k}\left\langle \boldsymbol{y}_{i-1} - \boldsymbol{v}, -(\boldsymbol{y}_i - \boldsymbol{y}_{i-1}) + \frac{1}{m}(\hat{\boldsymbol{y}}_i - \boldsymbol{y}_{i-1})\right\rangle \\ =\ & A_k(\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \tilde{\boldsymbol{y}}_k)) + \frac{1}{\gamma}\sum_{i=1}^{k}\langle \boldsymbol{y}_{i-1} - \boldsymbol{v}, -\check{\boldsymbol{v}}_i\rangle. \end{aligned} \tag{4.28}$$

where $\tilde{\boldsymbol{x}}_k = \frac{1}{A_k}\sum_{i=1}^{k} a_i\boldsymbol{x}_i$, $\tilde{\boldsymbol{y}}_k = \frac{1}{A_k}\sum_{i=1}^{k}(a_i\boldsymbol{y}_i + (m-1)a_i(\boldsymbol{y}_i - \boldsymbol{y}_{i-1}))$ (as defined in (4.33)) and the last equality is by the definition of the Lagrangian $\mathcal{L}(\cdot,\cdot)$ and $\{\check{\boldsymbol{v}}_i\}$ in Lemma 25.

Then by combining Eqs. (4.27)-(4.28) and Lemma 24, we have

$$\begin{aligned}
&A_k(\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \tilde{\boldsymbol{y}}_k)) \\
&\leq \left(\psi_k(\boldsymbol{y}_k) + \phi_k(\boldsymbol{x}_k) + ma_k \left\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}) \right\rangle \right) - \frac{1}{4\gamma}\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2 \\
&\quad - \frac{1}{4\gamma}\sum_{i=1}^{k}\|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2 + \frac{1}{\gamma}\sum_{i=1}^{k}\langle \boldsymbol{y}_{i-1}, \check{\boldsymbol{v}}_i\rangle - \frac{1}{\gamma}\sum_{i=1}^{k}\langle \boldsymbol{v}, \check{\boldsymbol{v}}_i\rangle \\
&\leq \Big(\frac{1}{2\gamma}\|\boldsymbol{v} - \boldsymbol{y}_0\|^2 - \frac{1}{2\gamma}\|\boldsymbol{v} - \boldsymbol{y}_k\|^2 + \frac{\gamma}{2}\|\boldsymbol{u} - \boldsymbol{x}_0\|^2 - \frac{\gamma + \sigma A_k}{2}\|\boldsymbol{u} - \boldsymbol{x}_k\|^2\Big) \\
&\qquad + ma_k \left\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}) \right\rangle - \frac{1}{4\gamma}\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2 - \frac{1}{4\gamma}\sum_{i=1}^{k}\|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2 \\
&\quad + \frac{1}{\gamma}\sum_{i=1}^{k}\langle \boldsymbol{y}_{i-1}, \check{\boldsymbol{v}}_i\rangle - \frac{1}{\gamma}\sum_{i=1}^{k}\langle \boldsymbol{v}, \check{\boldsymbol{v}}_i\rangle .
\end{aligned}$$

Finally, we have from the fact that $\boldsymbol{y}_k$ and $\boldsymbol{y}_{k-1}$ differ in only the S^{j_k} components, Young's inequality, and the definition of a_k that

$$\begin{aligned}
&|ma_k\langle \boldsymbol{x}_k - \boldsymbol{u}, \boldsymbol{A}^T(\boldsymbol{y}_k - \boldsymbol{y}_{k-1})\rangle| \\
&= |ma_k\ \langle \boldsymbol{A}^{S^{j_k}}(\boldsymbol{x}_k - \boldsymbol{u}), \boldsymbol{y}_k^{S^{j_k}} - \boldsymbol{y}_{k-1}^{S^{j_k}}\rangle| \\
&\leq ma_k\|\boldsymbol{A}^{S^{j_k}}(\boldsymbol{x}_k - \boldsymbol{u})\|\|\boldsymbol{y}_k^{S^{j_k}} - \boldsymbol{y}_{k-1}^{S^{j_k}}\| \\
&\leq (ma_k)^2\gamma\|\boldsymbol{A}^{S^{j_k}}(\boldsymbol{x}_k - \boldsymbol{u})\|^2 + \frac{1}{4\gamma}\|\boldsymbol{y}_k^{S^{j_k}} - \boldsymbol{y}_{k-1}^{S^{j_k}}\|^2 \\
&\leq (m\hat{L}a_k)^2\gamma\|\boldsymbol{x}_k - \boldsymbol{u}\|^2 + \frac{1}{4\gamma}\|\boldsymbol{y}_k^{S^{j_k}} - \boldsymbol{y}_{k-1}^{S^{j_k}}\|^2 \\
&= (m\hat{L}a_k)^2\gamma\|\boldsymbol{x}_k - \boldsymbol{u}\|^2 + \frac{1}{4\gamma}\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2 \\
&= \frac{\gamma + \sigma A_{k-1}}{4}\|\boldsymbol{x}_k - \boldsymbol{u}\|^2 + \frac{1}{4\gamma}\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2 \\
&\leq \frac{\gamma + \sigma A_k}{4}\|\boldsymbol{x}_k - \boldsymbol{u}\|^2 + \frac{1}{4\gamma}\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|^2
\end{aligned}$$

leading to

$$\begin{aligned}
&A_k(\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \tilde{\boldsymbol{y}}_k)) + \frac{1}{2\gamma}\|\boldsymbol{v} - \boldsymbol{y}_k\|^2 + \frac{\gamma + \sigma A_k}{4}\|\boldsymbol{u} - \boldsymbol{x}_k\|^2 \\
&\leq \frac{\gamma}{2}\|\boldsymbol{u} - \boldsymbol{x}_0\|^2 + \frac{1}{2\gamma}\|\boldsymbol{v} - \boldsymbol{y}_0\|^2 - \frac{1}{4\gamma}\sum_{i=1}^{k}\|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2 \\
&\quad + \frac{1}{\gamma}\sum_{i=1}^{k}\langle \boldsymbol{y}_{i-1}, \check{\boldsymbol{v}}_i\rangle - \frac{1}{\gamma}\sum_{i=1}^{k}\langle \boldsymbol{v}, \check{\boldsymbol{v}}_i\rangle
\end{aligned}$$

and completing the proof. □

Lemma 29. *Suppose that* $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ *is a Nash point for* (PD-GLP). *Then the iterates* $\boldsymbol{x}_k, \boldsymbol{y}_k$ *from Algorithm 8 satisfy*

$$\begin{aligned} &\mathbb{E}\Big[\frac{\gamma+\sigma A_k}{4}\|\boldsymbol{x}^*-\boldsymbol{x}_k\|^2+\frac{1}{2\gamma}\|\boldsymbol{y}^*-\boldsymbol{y}_k\|^2+\frac{1}{4\gamma}\sum_{i=1}^{k}\|\boldsymbol{y}_i-\boldsymbol{y}_{i-1}\|^2\Big] \\ &\leq \frac{\gamma}{2}\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2+\frac{1}{2\gamma}\|\boldsymbol{y}^*-\boldsymbol{y}_0\|^2, \end{aligned} \tag{4.29}$$

where the expectation is w.r.t. all the randomness in the algorithm.

Proof. Note that the existence of a Nash point is assumed in Assumption 10. With $(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{x}^*, \boldsymbol{y}^*)$, by the definition of Nash equilibrium, we have

$$\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{y}^*) - \mathcal{L}(\boldsymbol{x}^*, \tilde{\boldsymbol{y}}_k) = (\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{y}^*) - \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}^*)) - (\mathcal{L}(\boldsymbol{x}^*, \tilde{\boldsymbol{y}}_k) - \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}^*)) \geq 0. \tag{4.30}$$

By setting $(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ in the result of Lemma 28, using (4.30) to eliminate the first term on the left-hand side of the inequality, and rearranging, we obtain

$$\begin{aligned} &\frac{\gamma+\sigma A_k}{4}\|\boldsymbol{x}^*-\boldsymbol{x}_k\|^2+\frac{1}{2\gamma}\|\boldsymbol{y}^*-\boldsymbol{y}_k\|^2+\frac{1}{4\gamma}\sum_{i=1}^{k}\|\boldsymbol{y}_i-\boldsymbol{y}_{i-1}\|^2 \\ &\leq \frac{\gamma}{2}\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2+\frac{1}{2\gamma}\|\boldsymbol{y}^*-\boldsymbol{y}_0\|^2+\frac{1}{\gamma}\sum_{i=1}^{k}\langle \boldsymbol{y}_{i-1}, \check{\boldsymbol{v}}_i\rangle - \frac{1}{\gamma}\sum_{i=1}^{k}\langle \boldsymbol{v}, \check{\boldsymbol{v}}_i\rangle . \end{aligned}$$

The result will follow when we show that the expectation (with respect to all the randomness) of the last two terms on the right-hand side is zero. Since $\boldsymbol{y}_{i-1} \in \mathcal{F}_{i-1}$, we have

$$\mathbb{E}\left[\sum_{i=1}^{k}\langle \boldsymbol{y}_{i-1}, \check{\boldsymbol{v}}_i\rangle\right] = \mathbb{E}\left[\mathbb{E}\left[\sum_{i=1}^{k}\langle \boldsymbol{y}_{i-1}, \check{\boldsymbol{v}}_i\rangle \,\middle|\, \mathcal{F}_{i-1}\right]\right] = \mathbb{E}\left[\sum_{i=1}^{k}\langle \boldsymbol{y}_{i-1}, \mathbb{E}\left[\check{\boldsymbol{v}}_i|\mathcal{F}_{i-1}\right]\rangle\right] = 0, \tag{4.31}$$

which takes care of the second-last term. For any fixed $\boldsymbol{v} \in \mathbb{R}^n$, we also have

$$\mathbb{E}\Big[\frac{1}{\gamma}\sum_{i=1}^{k}\langle \boldsymbol{v}, \check{\boldsymbol{v}}_i\rangle\Big] = 0, \tag{4.32}$$

which takes care of the last term, and completes the proof. □

Next we state a technical lemma, proved in an earlier paper, which provides the final ingredient for the proof of Theorem 8.

Lemma 30. *[[SWD21]] Let* $\{A_k\}_{k\geq 0}$ *be a sequence of nonnegative real numbers such that* $A_0 = 0$ *and* A_k *is defined recursively via* $A_k = A_{k-1} + \sqrt{c_1^2 + c_2 A_{k-1}}$, *where* $c_1 > 0$, *and* $c_2 \geq 0$. *Define* $K_0 = \lceil \frac{c_2}{9c_1} \rceil$. *Then*

$$A_k \geq \begin{cases} \frac{c_2}{9}\Big(k - K_0 + \max\Big\{3\sqrt{\frac{c_1}{c_2}}, 1\Big\}\Big)^2, & \text{if } c_2 > 0 \text{ and } k > K_0, \\ c_1 k, & \text{otherwise.} \end{cases}$$

We are now ready to prove our main result. Theorem 8 provides the convergence results for Algorithm 8. Note that γ in the theorem (as in the algorithm) is a positive parameter that can be tuned.

Theorem 8. *Let $\boldsymbol{x}_k$, $\boldsymbol{y}_k$, $k \in [K]$, be the iterates of Algorithm 8 and let $\tilde{\boldsymbol{x}}_k$, $\tilde{\boldsymbol{y}}_k$ be defined by*

$$\tilde{\boldsymbol{x}}_k = \frac{1}{A_k}\sum_{i=1}^{k} a_i \boldsymbol{x}_i, \quad \tilde{\boldsymbol{y}}_k = \frac{1}{A_k}\sum_{i=1}^{k}(a_i \boldsymbol{y}_i + (m-1)a_i(\boldsymbol{y}_i - \boldsymbol{y}_{i-1})), \tag{4.33}$$

for $k \in [K]$. Let $\mathcal{W}_k \subset \mathcal{X} \times \mathbb{R}^n$, $k \in [K]$, be a sequence of compact convex sets such that $(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{y}}_k) \in \mathcal{W}_k \subset \mathcal{W} \subset \mathcal{X} \times \mathbb{R}^n$, where $\mathcal{W}$ is also convex and compact. Then:

$$\begin{aligned} &\mathbb{E}\Big[\sup_{(\boldsymbol{u},\boldsymbol{v})\in\mathcal{W}_k}\{\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \tilde{\boldsymbol{y}}_k)\}\Big] \\ &\leq \frac{1}{A_k}\Big(\mathbb{E}\Big[\frac{\gamma}{2}\|\hat{\boldsymbol{u}} - \boldsymbol{x}_0\|^2 + \frac{1}{\gamma}\|\hat{\boldsymbol{v}} - \boldsymbol{y}_0\|^2\Big] + \frac{\gamma}{2}\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2 + \frac{1}{2\gamma}\|\boldsymbol{y}^* - \boldsymbol{y}_0\|^2\Big), \end{aligned} \tag{4.34}$$

where $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}) = \arg\sup_{(\boldsymbol{u},\boldsymbol{v})\in\mathcal{W}_k}\{\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \tilde{\boldsymbol{y}}_k)\}$. Furthermore,

$$\mathbb{E}\Big[\frac{\gamma + \sigma A_k}{4}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 + \frac{1}{2\gamma}\|\boldsymbol{y}_k - \boldsymbol{y}^*\|^2\Big] \leq \frac{\gamma}{2}\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2 + \frac{1}{2\gamma}\|\boldsymbol{y}^* - \boldsymbol{y}_0\|^2. \tag{4.35}$$

Define $K_0 = \lceil \frac{\sigma}{18\hat{L}m\gamma} \rceil$. Then in the bounds above:

$$A_k \geq \max\left\{\frac{k}{2\hat{L}m}, \frac{\sigma}{(6\hat{L}m)^2\gamma}\Big(k - K_0 + \max\Big\{3\sqrt{2\hat{L}m\gamma/\sigma}, 1\Big\}\Big)^2\right\}.$$

Proof. To provide a guarantee in terms of primal-dual gap, we need to take the supremum on $\boldsymbol{u}, \boldsymbol{v}$ of $\{\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \tilde{\boldsymbol{y}}_k))\}$ over $\mathcal{W}_k$; we denote the arg sup by $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})$. When we subsequently take the expectation, we have to account for the fact that $\hat{\boldsymbol{v}}$ will be correlated with the randomness in the iteration history. We can however, use Lemmas 25 and 29 to give the upper bound on $\mathbb{E}\Big[-\sum_{i=1}^{k}\langle \hat{\boldsymbol{v}}, \check{\boldsymbol{v}}_i\rangle\Big]$.

From (4.26), using the fact that $\frac{1}{2\gamma}\|\boldsymbol{v}-\boldsymbol{y}_k\|^2+\frac{\gamma+\sigma A_k}{4}\|\boldsymbol{u}-\boldsymbol{x}_k\|^2\geq 0$, we have

$$
\begin{aligned}
&\mathbb{E}\Big[A_k \sup_{(\boldsymbol{u},\boldsymbol{v})\in\mathcal{W}_k}\{\mathcal{L}(\tilde{\boldsymbol{x}}_k,\boldsymbol{v})-\mathcal{L}(\boldsymbol{u},\tilde{\boldsymbol{y}}_k)\}\Big]\\
&\leq \mathbb{E}\Big[\frac{\gamma}{2}\|\hat{\boldsymbol{u}}-\boldsymbol{x}_0\|^2+\frac{1}{2\gamma}\|\boldsymbol{y}_0-\hat{\boldsymbol{v}}\|^2\Big]-\frac{1}{4\gamma}\mathbb{E}\Big[\sum_{i=1}^k\|\boldsymbol{y}_i-\boldsymbol{y}_{i-1}\|^2\Big]+\frac{1}{\gamma}\mathbb{E}\Big[\sum_{i=1}^k\langle\boldsymbol{y}_{i-1},\check{\boldsymbol{v}}_i\rangle\Big]\\
&\quad+\frac{1}{\gamma}\mathbb{E}\Big[-\sum_{i=1}^k\langle\hat{\boldsymbol{v}},\check{\boldsymbol{v}}_i\rangle\Big]\\
&\leq \mathbb{E}\Big[\frac{\gamma}{2}\|\hat{\boldsymbol{u}}-\boldsymbol{x}_0\|^2+\frac{1}{2\gamma}\|\boldsymbol{y}_0-\hat{\boldsymbol{v}}\|^2\Big]-\frac{1}{4\gamma}\mathbb{E}\Big[\sum_{i=1}^k\|\boldsymbol{y}_i-\boldsymbol{y}_{i-1}\|^2\Big]\\
&\quad+\frac{1}{\gamma}\mathbb{E}\Big[\frac{1}{2}\|\boldsymbol{y}_0-\hat{\boldsymbol{v}}\|^2+\frac{1}{2}\sum_{i=1}^k\|\boldsymbol{y}_i-\boldsymbol{y}_{i-1}\|^2\Big]\\
&= \mathbb{E}\Big[\frac{\gamma}{2}\|\hat{\boldsymbol{u}}-\boldsymbol{x}_0\|^2+\frac{1}{\gamma}\|\boldsymbol{y}_0-\hat{\boldsymbol{v}}\|^2\Big]+\frac{1}{4\gamma}\mathbb{E}\Big[\sum_{i=1}^k\|\boldsymbol{y}_i-\boldsymbol{y}_{i-1}\|^2\Big]\\
&\leq \mathbb{E}\Big[\frac{\gamma}{2}\|\hat{\boldsymbol{u}}-\boldsymbol{x}_0\|^2+\frac{1}{\gamma}\|\boldsymbol{y}_0-\hat{\boldsymbol{v}}\|^2\Big]+\frac{\gamma}{2}\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2+\frac{1}{2\gamma}\|\boldsymbol{y}^*-\boldsymbol{y}_0\|^2. \qquad (4.36)
\end{aligned}
$$

where the first inequality is by Eq. (4.26), the second inequality is by Lemma 25 and (4.31), and the last inequality is by Lemma 29. This proves the first claim (4.34). The second claim (4.35) follows from Lemma 29 with the fact that $\frac{1}{4\gamma}\mathbb{E}\Big[\sum_{i=1}^k\|\boldsymbol{y}_i-\boldsymbol{y}_{i-1}\|^2\Big]\geq 0$. The final claim concerning A_k follows from Lemma 30 when we set

$$c_1=\frac{1}{2\hat{L}m},\quad c_2=\frac{\sigma}{(2\hat{L}m)^2\gamma}.$$

□

Observe that $(\hat{\boldsymbol{u}},\hat{\boldsymbol{v}})$ in the theorem statement exists because of compactness of $\mathcal{W}_k$ and our assumptions on $r(\cdot)$. The parameter γ can be tuned to balance the relative weights of primal and dual initial quantities $\|\boldsymbol{x}^*-\boldsymbol{x}_0\|$ and $\|\boldsymbol{y}^*-\boldsymbol{y}_0\|$ (or estimates of these quantities), which can significantly influence practical performance of the method.

In addition to the guarantee on the variational form, due to the linear structure, we also provide explicit guarantees for both the objective and the constraints in (GLP), stated in the following corollary.

Corollary 1. *In Algorithm 8, for all $k\geq 1$, $\tilde{\boldsymbol{x}}_k$ satisfies*

$$
\begin{aligned}
\mathbb{E}[\|\boldsymbol{y}^*\|\cdot\|\boldsymbol{A}\tilde{\boldsymbol{x}}_k-\boldsymbol{b}\|] &\leq \frac{\gamma\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2+\frac{1}{2\gamma}\|\boldsymbol{y}^*-\boldsymbol{y}_0\|^2+\frac{1}{\gamma}\mathbb{E}[\|\boldsymbol{v}-\boldsymbol{y}_0\|^2]}{A_k},\\
|\mathbb{E}[(\boldsymbol{c}^T\tilde{\boldsymbol{x}}_k+r(\tilde{\boldsymbol{x}}_k))-(\boldsymbol{c}^T\boldsymbol{x}^*+r(\boldsymbol{x}^*))]| &\leq \frac{\gamma\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2+\frac{1}{2\gamma}\|\boldsymbol{y}^*-\boldsymbol{y}_0\|^2+\frac{1}{\gamma}\mathbb{E}[\|\boldsymbol{v}-\boldsymbol{y}_0\|^2]}{A_k},
\end{aligned}
$$

where $\boldsymbol{v} = 2\frac{\|\boldsymbol{y}^*\|}{\|\boldsymbol{A}\tilde{\boldsymbol{x}}_k - \boldsymbol{b}\|}(\boldsymbol{A}\tilde{\boldsymbol{x}}_k - \boldsymbol{b})$.

Proof. Assume that $\|\boldsymbol{A}\tilde{\boldsymbol{x}}_k - \boldsymbol{b}\| \neq 0$, as otherwise the first bound follows trivially. Let $\boldsymbol{u} = \boldsymbol{x}^*$ and $\boldsymbol{v} = \frac{2\|\boldsymbol{y}^*\|(\boldsymbol{A}\tilde{\boldsymbol{x}}_k - \boldsymbol{b})}{\|\boldsymbol{A}\tilde{\boldsymbol{x}}_k - \boldsymbol{b}\|}$. Then we have

$$\begin{aligned}
&\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \tilde{\boldsymbol{y}}_k) \\
&= (\boldsymbol{c}^T\tilde{\boldsymbol{x}}_k + r(\tilde{\boldsymbol{x}}_k) + 2\|\boldsymbol{y}^*\|\|\boldsymbol{A}\tilde{\boldsymbol{x}}_k - \boldsymbol{b}\|) - (\boldsymbol{c}^T\boldsymbol{x}^* + r(\boldsymbol{x}^*) + \tilde{\boldsymbol{y}}_k^T(\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b})) \\
&= (\boldsymbol{c}^T(\tilde{\boldsymbol{x}}_k - \boldsymbol{x}^*) + r(\tilde{\boldsymbol{x}}_k) - r(\boldsymbol{x}^*)) + 2\|\boldsymbol{y}^*\|\|\boldsymbol{A}\tilde{\boldsymbol{x}}_k - \boldsymbol{b}\|. \qquad (4.37)
\end{aligned}$$

For any fixed $\boldsymbol{u}$, and any $\boldsymbol{v} \in \mathbb{R}^n$ possibly depending on the randomness in the algorithm, we have from Lemma 28, taking expectation over all the randomness, that

$$\begin{aligned}
&A_k\mathbb{E}[\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \tilde{\boldsymbol{y}}_k)] \\
&\leq \mathbb{E}\left[A_k(\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \tilde{\boldsymbol{y}}_k)) + \frac{1}{2\gamma}\|\boldsymbol{v} - \boldsymbol{y}_k\|^2 + \frac{\gamma + \sigma A_k}{4}\|\boldsymbol{u} - \boldsymbol{x}_k\|^2\right] \\
&\leq \frac{\gamma}{2}\|\boldsymbol{u} - \boldsymbol{x}_0\|^2 + \frac{1}{2\gamma}\mathbb{E}[\|\boldsymbol{v} - \boldsymbol{y}_0\|^2] - \mathbb{E}\Big[\frac{1}{4\gamma}\sum_{i=1}^k\|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2\Big] \\
&\quad + \frac{1}{\gamma}\mathbb{E}\Big[\sum_{i=1}^k\langle\boldsymbol{y}_{i-1}, \check{\boldsymbol{v}}_i\rangle\Big] - \frac{1}{\gamma}\mathbb{E}\Big[\sum_{i=1}^k\langle\boldsymbol{v}, \check{\boldsymbol{v}}_i\rangle\Big]. \qquad (4.38)
\end{aligned}$$

Meanwhile, we have

$$\begin{aligned}
&-\frac{1}{4\gamma}\mathbb{E}\Big[\sum_{i=1}^k\|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2\Big] - \frac{1}{\gamma}\mathbb{E}\Big[\sum_{i=1}^k\langle\boldsymbol{v}, \check{\boldsymbol{v}}_i\rangle\Big] \\
&\leq \frac{1}{2\gamma}\mathbb{E}\Big[\|\boldsymbol{y}_0 - \boldsymbol{v}\|^2\Big] + \frac{1}{4\gamma}\mathbb{E}\Big[\sum_{i=1}^k\|\boldsymbol{y}_i - \boldsymbol{y}_{i-1}\|^2\Big] \\
&\leq \frac{1}{2\gamma}\mathbb{E}[\|\boldsymbol{y}_0 - \boldsymbol{v}\|^2] + \frac{\gamma}{2}\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2 + \frac{1}{2\gamma}\|\boldsymbol{y}^* - \boldsymbol{y}_0\|^2,
\end{aligned}$$

where the first inequality is by Lemma 25 and the second inequality is by Lemma 29. Since $\boldsymbol{y}_{i-1} \in \mathcal{F}_{i-1}$, we have as in (4.31) that

$$\frac{1}{\gamma}\mathbb{E}\Big[\sum_{i=1}^k\langle\boldsymbol{y}_{i-1}, \check{\boldsymbol{v}}_i\rangle\Big] = 0. \qquad (4.39)$$

By combining Eq. (4.37)-(4.39) with $\boldsymbol{u} = \boldsymbol{x}^*$, we have

$$\begin{aligned}
&\mathbb{E}[(\boldsymbol{c}^T(\tilde{\boldsymbol{x}}_k - \boldsymbol{x}^*) + r(\tilde{\boldsymbol{x}}_k) - r(\boldsymbol{x}^*)) + 2\|\boldsymbol{y}^*\|\|\boldsymbol{A}\tilde{\boldsymbol{x}}_k - \boldsymbol{b}\|] \\
&\leq \frac{\gamma\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2 + \frac{1}{\gamma}\mathbb{E}[\|\boldsymbol{v} - \boldsymbol{y}_0\|^2] + \frac{1}{2\gamma}\|\boldsymbol{y}^* - \boldsymbol{y}_0\|^2}{A_k}. \qquad (4.40)
\end{aligned}$$

By the KKT condition of (PD-GLP) and the optimality of $(\boldsymbol{x}^*, \boldsymbol{y}^*)$, we have for all $\boldsymbol{x} \in \mathcal{X}$ that

$$(\boldsymbol{c}^T\boldsymbol{x} + r(\boldsymbol{x})) - (\boldsymbol{c}^T\boldsymbol{x}^* + r(\boldsymbol{x}^*)) - \langle \boldsymbol{y}^*, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\rangle \geq 0, \tag{4.41}$$

and thus

$$(\boldsymbol{c}^T\boldsymbol{x} + r(\boldsymbol{x})) - (\boldsymbol{c}^T\boldsymbol{x}^* + r(\boldsymbol{x}^*)) \geq -\|\boldsymbol{y}^*\|\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|. \tag{4.42}$$

By combining Eq. (4.40) and Eq. (4.42), we have

$$\mathbb{E}[\|\boldsymbol{y}^*\| \cdot \|\boldsymbol{A}\tilde{\boldsymbol{x}}_k - \boldsymbol{b}\|] \leq \frac{\gamma\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2 + \frac{1}{\gamma}\mathbb{E}[\|\boldsymbol{v} - \boldsymbol{y}_0\|^2] + \frac{1}{2\gamma}\|\boldsymbol{y}^* - \boldsymbol{y}_0\|^2}{A_k}, \tag{4.43}$$

proving our first claim. The second claim is obtained by combining Eqs. (4.40), (4.42), and (4.43). □

In CLVR, we allow for arbitrary $(\boldsymbol{x}_0, \boldsymbol{y}_0) \in \mathcal{X} \times \mathbb{R}^n$. Nevertheless, by setting $\boldsymbol{y} = \boldsymbol{0}_n$, we can obtain $\boldsymbol{z}_0 = \boldsymbol{0}_d$ at no cost — a useful strategy for large-scale problems since it avoids the (potentially expensive) single matrix-vector multiplication w.r.t. $\boldsymbol{A}$.

Remark 1. *For our algorithm, the condition that $\boldsymbol{y}$ belongs to a whole Euclidean space is key to proving our convergence result. Although many convex-concave min-max problems have constraints or nonsmooth regularizers on the dual variable $\boldsymbol{y}$, we can still reformulate it so that the dual variable is unconstrained and no nonsmooth regularizers are applied to it. For instance, we can verify that*

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathcal{Y}} L(\boldsymbol{x}, \boldsymbol{y}) \equiv \min_{\boldsymbol{x}\in\mathcal{X}, \boldsymbol{z}\mathrm{dom}(\delta^*_{\mathcal{Y}})} \max_{\boldsymbol{y}\in\mathbb{R}^n} L(\boldsymbol{x}, \boldsymbol{y}) - \langle \boldsymbol{z}, \boldsymbol{y}\rangle + \delta^*_{\mathcal{Y}}(\boldsymbol{z}), \tag{4.44}$$

where both $\mathcal{X}$ and $\mathcal{Y}$ are convex sets with $\mathcal{Y} \in \mathbb{R}^n$, $\delta^*_{\mathcal{Y}}(z) = \sup_{\boldsymbol{y}\in\mathcal{Y}} \boldsymbol{y}^T\boldsymbol{z}$.

4.4.2 Lazy update for sparse and structured GLP

In Algorithm 8, direct computation of the iterates $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ and the output points $(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{y}}_k)$ can be expensive. However, [DL15] showed that it is possible to only update the averaged vector in the coordinate block chosen for that iteration. This strategy requires us to record the most recent update for each coordinate block and update it only when it is selected again, which is tricky and needs to be implemented carefully. For sparse and block coordinate-separable instances of (PD-GLP), we show that by introducing auxiliary variables that are sparsely connected, we can significantly simplify CLVR and make its complexity scale independently of the ambient dimension $n \cdot d$, instead scaling with $\mathrm{nnz}(\boldsymbol{A})$.

Lazification. In Algorithm 8, for dense $\boldsymbol{A}$, the $O(|S^{j_k}|d)$ cost of Steps 6 and 8 dominates the $O(d)$ cost of Steps 4 and 9. However, when $\boldsymbol{A}$ is sparse and $|S^{j_k}|$ is small, the cost of Steps 6 and 8 will be $O(\text{nnz}(\boldsymbol{A}^{S^{j_k}}))$, which may be less than the $O(d)$ cost of Steps 4 and 9. Using this observation, we show that the nature of the dual averaging update enables us to propose an efficient implementation whose complexity depends on $\text{nnz}(\boldsymbol{A})$ rather than $n \cdot d$.

Recall that we partition $[n]$ into subsets $\{S^1, S^2, \ldots, S^m\}$ and use $\boldsymbol{A}^{S^j}(j \in [m])$ to denote the jth row block of $\boldsymbol{A}$. For each block $\boldsymbol{A}^{S^j}$, we use $C^j \subset [d]$ to denote the indices of those columns of $\boldsymbol{A}^{S^j}$ that contain at least one nonzero element. (Of course, $\{C^1, \ldots, C^m\}$ is not in general a partition of $[d]$ as different subsets S^j may have non-zeros in the same columns.) We assume further that $\mathcal{X}$ and r are coordinate separable, that is, $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$ with $\mathcal{X}_i \subset \mathbb{R}$ for all i and $r(\boldsymbol{x}) := \sum_{i=1}^d r(\boldsymbol{x}^i)$ with $\boldsymbol{x}^i \in \mathcal{X}_i$.

In Step 4 of Algorithm 8, the separability of $\mathcal{X}$ and r means that an update to one coordinate block of $\boldsymbol{x}$ — say the C^{j_k} block in the update from $\boldsymbol{x}_{k-1}$ to $\boldsymbol{x}_k$, which requires a projection and an application of the proximal operator — does not influence other coordinates $\boldsymbol{x}_k^i$ for $i \notin C^{j_k}$. Moreover, we can efficiently maintain an implicit representation of $\boldsymbol{q}_k$, in terms of a newly introduced auxiliary vector $\boldsymbol{r}_k$; see Lemma 31 below. Similarly, to update $\tilde{\boldsymbol{y}}_k$ efficiently, we maintain an implicit representation of $\tilde{\boldsymbol{y}}_k$ via an auxiliary vector $\boldsymbol{s}_k$, as shown in Lemma 32 below.

It is generally not possible to maintain $\tilde{\boldsymbol{x}}_k$ efficiently since, in principle, all components of $\boldsymbol{x}_k$ can change on every iteration, and we wish to avoid the $O(d)$ cost of evaluating every full $\boldsymbol{x}_k$. We seek instead to output a proxy $\hat{\boldsymbol{x}}_K$ for $\tilde{\boldsymbol{x}}_K$ from Algorithm 8 such that $\mathbb{E}[\hat{\boldsymbol{x}}_K] = \tilde{\boldsymbol{x}}_K$. One possible choice is to pick an index $k' \in [K]$ from the weighted discrete distribution $(\frac{a_1}{A_K}, \frac{a_2}{A_K}, \ldots, \frac{a_K}{A_K})$ (computing the scalar quantities a_k and A_k for $k \in [K]$ in advance), then setting $\hat{\boldsymbol{x}}_K = \boldsymbol{x}_{k'}$. A slightly more sophisticated strategy is to sample a predetermined number $\widehat{K}$ of vectors $\boldsymbol{x}_k$, $k \in [K]$, and define $\hat{\boldsymbol{x}}_K$ to be the simple average of these vectors. Once again, the indices are chosen from the weighted discrete distribution $(\frac{a_1}{A_K}, \frac{a_2}{A_K}, \ldots, \frac{a_K}{A_K})$. Note that the total expected cost of the K iterations of the algorithm (excluding the full-vector updates) is $O(K\text{nnz}(\boldsymbol{A})/m)$, while the total cost of evaluating the $\widehat{K}$ full vectors $\boldsymbol{x}_k$ and accumulating them into $\hat{\boldsymbol{x}}_K$ is $O(\widehat{K}d)$. Thus, for the full-step iterations not to dominate the total cost, we can choose $\widehat{K}$ to be $O(K\text{nnz}(\boldsymbol{A})/(md))$. Finally, we note that Theorem 8 is for $\tilde{\boldsymbol{x}}_k$, not $\hat{\boldsymbol{x}}_K$. However, since $\mathbb{E}[\hat{\boldsymbol{x}}_k] = \tilde{\boldsymbol{x}}_k$, we expect the convergence rates from the theorem to hold for $\hat{\boldsymbol{x}}_K$ in practice.

An implementation of Algorithm 8 that exploits the form of $\boldsymbol{q}_k$ in Lemma 31 and $\tilde{\boldsymbol{y}}_k$ in Lemma 32, and evaluates explicitly only those components of $\boldsymbol{x}_k$ needed to perform the rest of the iteration is given as Algorithm 10. This version also incorporates the strategy for obtaining $\hat{\boldsymbol{x}}_K$ by sampling $\widehat{K}$ iterates on which to evaluate the full vector $\boldsymbol{x}_k$.

Due to the efficient implementation in Algorithm 10, to attain an ϵ-accurate solution in terms of the primal-dual gap in Theorem 8, we need $O(\frac{\mathrm{nnz}(\boldsymbol{A})\hat{L}}{\epsilon})$ FLOPS, which corresponds to $O(\frac{\hat{L}}{\epsilon})$ data passes. As a result, because a smaller batch size leads to a smaller $\hat{L}$, we attain the best performance in terms of number of data passes when the batch size is set to one. However, as modern computer architecture has particular design for vectorized operations, lower runtime is obtained for a small batch size strictly larger than one (see Section 4.6).

Remark 2. *While we consider the case of fully coordinate separable r and $\mathcal{X}$ for simplicity, our lazy update approach is also applicable to the coordinate block partitioning case in which $\mathcal{X} = \mathcal{X}_1 \times \cdots \mathcal{X}_m$ with $\mathcal{X}_i \in \mathbb{R}^{d_i}$ $(i \in [m], \sum_{i=1}^m d_i = d)$ and $r(\boldsymbol{x}) := \sum_{i=1}^m r(\boldsymbol{x}^i)$ with $\boldsymbol{x}^i \in \mathcal{X}_i$. The difference is that for each coordinate block $\boldsymbol{A}^{S^j}(j \in [m])$, we overload $C^j \subset [m]$ to denote the set of blocks in $\boldsymbol{A}^{S^j}$ where each coordinate block contains at least one nonzero element.*

Remark 3. *Dual averaging has been shown to have significant advantage in producing sparser iterates than mirror descent in the context of online learning [Xia10, LW12]. It further leads to better bounds in well-conditioned finite-sum optimization [SJM20]. In this work, we show that dual averaging offers better flexibility with sparse matrices than does mirror descent.*

We need to compute explicitly only those components of $\boldsymbol{q}_{k-1}$ and $\boldsymbol{x}_k$ that are needed to update $\boldsymbol{y}_k$.

Lemma 31. *For $\{\boldsymbol{q}_k\}$ defined in Algorithm 8, we have*

$$\boldsymbol{q}_k = A_{k+1}(\boldsymbol{c} + \boldsymbol{z}_k) + \boldsymbol{r}_k, \quad k = 0, 1, 2, \ldots,$$

where $\boldsymbol{r}_0 = 0$ and for all $k = 1, 2, \ldots$

$$\boldsymbol{r}_k = \boldsymbol{r}_{k-1} + (ma_k - A_k)(\boldsymbol{z}_k - \boldsymbol{z}_{k-1}).$$

Proof. The proof is by induction. For $k = 0$, we have $\boldsymbol{q}_0 = A_1(\boldsymbol{c} + \boldsymbol{z}_0)$ which is true by definition. Assuming that the result holds for some index k, we show that it continues to hold for $k + 1$. We have from Step 9 of Algorithm 8, then using the inductive assumption, that

$$\begin{aligned}
\boldsymbol{q}_{k+1} &= a_{k+2}(\boldsymbol{z}_{k+1} + \boldsymbol{c}) + ma_{k+1}(\boldsymbol{z}_{k+1} - \boldsymbol{z}_k) + \boldsymbol{q}_k \\
&= a_{k+2}(\boldsymbol{z}_{k+1} + \boldsymbol{c}) + ma_{k+1}(\boldsymbol{z}_{k+1} - \boldsymbol{z}_k) + A_{k+1}(\boldsymbol{c} + \boldsymbol{z}_k) + \boldsymbol{r}_k \\
&= A_{k+2}\boldsymbol{c} + (A_{k+2} - A_{k+1})\boldsymbol{z}_{k+1} + ma_{k+1}(\boldsymbol{z}_{k+1} - \boldsymbol{z}_k) + A_{k+1}\boldsymbol{z}_k + \boldsymbol{r}_k \\
&= A_{k+2}(\boldsymbol{c} + \boldsymbol{z}_{k+1}) + (ma_{k+1} - A_{k+1})(\boldsymbol{z}_{k+1} - \boldsymbol{z}_k) + \boldsymbol{r}_k \\
&= A_{k+2}(\boldsymbol{c} + \boldsymbol{z}_{k+1}) + \boldsymbol{r}_{k+1},
\end{aligned}$$

as required. □

This lemma indicates that we can reconstruct $\boldsymbol{q}_k$ (or any subvector of $\boldsymbol{q}_k$ that we need, on demand), provided we maintain $\boldsymbol{z}_k$ and $\boldsymbol{r}_k$. Note that the update from $\boldsymbol{z}_k$ to $\boldsymbol{z}_{k+1}$ is sparse; these two vectors differ only in the components corresponding to the block C^{j_k}. To obtain $\boldsymbol{r}_{k+1}$ from $\boldsymbol{r}_k$, we need to add a scalar times the sparse vector $\boldsymbol{z}_{k+1} - \boldsymbol{z}_k$, so this update is also efficient.

We can also maintain an implicit representation of the averaged vector $\tilde{\boldsymbol{y}}_k$ efficiently, as shown in the following lemma.

Lemma 32. *For* $\{\tilde{\boldsymbol{y}}_k\}$ *defined in* (4.33)*, we have*

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{y}_k + \frac{1}{A_k}\boldsymbol{s}_k, \quad k = 1, 2, \ldots,$$

where $\boldsymbol{s}_0 = \boldsymbol{0}$ *and for all* $k = 1, 2, \ldots,$

$$\boldsymbol{s}_k = \boldsymbol{s}_{k-1} + ((m-1)a_k - A_{k-1})(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}).$$

Proof. Recall that $\tilde{\boldsymbol{y}}_K$ ($K \geq 1$) is defined in Step 11 of Algorithm 8. The proof is by induction. For $k = 1$, $\tilde{\boldsymbol{y}}_1 = \boldsymbol{y}_1 + \frac{1}{A_1}(m-1)a_1(\boldsymbol{y}_1 - \boldsymbol{y}_0) = \boldsymbol{y}_1 + \frac{1}{A_1}\boldsymbol{s}_1$ which is true by the definition of $\boldsymbol{s}_1$ and A_0. Next, we assume that the result holds for some $k \geq 2$ and show that it continues to hold for $k+1$. We have

$$\begin{aligned}
A_{k+1}\tilde{\boldsymbol{y}}_{k+1} &= \sum_{i=1}^{k+1}(a_i\boldsymbol{y}_i + (m-1)a_i(\boldsymbol{y}_i - \boldsymbol{y}_{i-1})) \\
&= A_k\tilde{\boldsymbol{y}}_k + a_{k+1}\boldsymbol{y}_{k+1} + (m-1)a_{k+1}(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k) \\
&= A_k\boldsymbol{y}_k + \boldsymbol{s}_k + a_{k+1}\boldsymbol{y}_{k+1} + (m-1)a_{k+1}(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k) \\
&= A_k(\boldsymbol{y}_k - \boldsymbol{y}_{k+1} + \boldsymbol{y}_{k+1}) + \boldsymbol{s}_k + a_{k+1}\boldsymbol{y}_{k+1} + (m-1)a_{k+1}(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k) \\
&= A_{k+1}\boldsymbol{y}_{k+1} + \boldsymbol{s}_k + ((m-1)a_{k+1} - A_k)(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k) \\
&= A_{k+1}\boldsymbol{y}_{k+1} + \boldsymbol{s}_{k+1}.
\end{aligned}$$

as required. □

4.4.3 Restart scheme

We now propose a fixed restart strategy with a fixed number of iterations per each restart epoch and discuss an adaptive restart strategy for the special case of standard-form LP, which corresponds to (GLP) with $r(\boldsymbol{x}) \equiv 0$ and $\mathcal{X} = \{\boldsymbol{x} : x_i \geq 0, i \in [d]\}$. We write

$$\min_{\boldsymbol{x}} \boldsymbol{c}^T\boldsymbol{x} \text{ s. t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq \boldsymbol{0}_d, \tag{LP}$$

Algorithm 10 Coordinate Linear Variance Reduction with Lazy Update (Lazy CLVR)

Input: $\tilde{\boldsymbol{x}}_0 = \boldsymbol{x}_0 = \boldsymbol{x}_{-1} \in \mathcal{X}, \boldsymbol{y}_0 \in \mathbb{R}^n, \boldsymbol{z}_0 = \boldsymbol{A}^T\boldsymbol{y}_0$, $\gamma > 0$, $\hat{L} > 0$, K, $\widehat{K}$, m, $\{S^1, S^2, \ldots, S^m\}$, $\{C^1, C^2, \ldots, C^m\}$.
$a_0 = A_0 = 0, a_1 = A_1 = \frac{1}{2\hat{L}m}, \boldsymbol{q}_0 = a_1(\boldsymbol{z}_0 + \boldsymbol{c})$, $\boldsymbol{r}_0 = \boldsymbol{0}_d$, $\boldsymbol{s}_0 = \boldsymbol{0}_n$.
for $k = 1, 2, \ldots, K-1$ **do**
  $a_{k+1} = \frac{\sqrt{1+\sigma A_k/\gamma}}{2\hat{L}m}$, $A_{k+1} = A_k + a_{k+1}$.
end for
Choose indices $\{\ell_1, \ldots, \ell_{\widehat{K}}\}$ i.i.d. from $[K]$ according to the distribution $\left\{\frac{a_1}{A_K}, \frac{a_2}{A_K}, \ldots, \frac{a_K}{A_K}\right\}$.
for $k = 1, 2, \ldots, K$ **do**
  Pick j_k uniformly at random in $[m]$.
  if $k = \ell_i$ for some $i = 1, 2, \ldots, \widehat{K}$ **then**
    $\boldsymbol{q}_{k-1} = A_k(\boldsymbol{c} + \boldsymbol{z}_{k-1}) + \boldsymbol{r}_{k-1}$.
    $\boldsymbol{x}_k = \mathrm{prox}_{\frac{1}{\gamma}A_k r}(\boldsymbol{x}_0 - \frac{1}{\gamma}\boldsymbol{q}_{k-1})$.
  else
    $\boldsymbol{q}_{k-1}^{C^{j_k}} = A_k(\boldsymbol{c}^{C^{j_k}} + \boldsymbol{z}_{k-1}^{C^{j_k}}) + \boldsymbol{r}_{k-1}^{C^{j_k}}$.
    $\boldsymbol{x}_k^{C^{j_k}} = \mathrm{prox}_{\frac{1}{\gamma}A_k r}(\boldsymbol{x}_0^{C^{j_k}} - \frac{1}{\gamma}\boldsymbol{q}_{k-1}^{C^{j_k}})$.
  end if
  $\boldsymbol{y}_k^{S^{j_k}} = \boldsymbol{y}_{k-1}^{S^{j_k}} + \gamma m a_k(\boldsymbol{A}^{S^{j_k}, C^{j_k}}\boldsymbol{x}_k^{C^{j_k}} - \boldsymbol{b}^{S^{j_k}}), \boldsymbol{y}_{k-1}^{S^i}$ for all $i \neq j_k$.
  $\boldsymbol{z}_k^{C^{j_k}} = \boldsymbol{z}_{k-1}^{C^{j_k}} + (\boldsymbol{A}^{S^{j_k}, C^{j_k}})^T(\boldsymbol{y}_k^{S^{j_k}} - \boldsymbol{y}_{k-1}^{S^{j_k}})$, $\boldsymbol{z}_k^i = \boldsymbol{z}_{k-1}^i$ for all $i \notin C^{j_k}$;
  $\boldsymbol{r}_k^{C^{j_k}} = \boldsymbol{r}_{k-1}^{C^{j_k}} + (m a_k - A_k)(\boldsymbol{z}_k^{C^{j_k}} - \boldsymbol{z}_{k-1}^{C^{j_k}})$, $\boldsymbol{r}_k^i = \boldsymbol{r}_{k-1}^i$ for all $i \notin C^{j_k}$;
  $\boldsymbol{s}_k^{S^{j_k}} = \boldsymbol{s}_{k-1}^{S^{j_k}} + ((m-1)a_k - A_{k-1})(\boldsymbol{y}_k^{S^{j_k}} - \boldsymbol{y}_{k-1}^{S^{j_k}})$, $\boldsymbol{s}_k^i = \boldsymbol{s}_{k-1}^i$ for all $i \notin S^{j_k}$;
end for
$\hat{\boldsymbol{x}}_K = \frac{1}{\widehat{K}}\sum_{i=1}^{\widehat{K}} \boldsymbol{x}_{\ell_i}$.
$\tilde{\boldsymbol{y}}_K = \boldsymbol{y}_K + \frac{1}{A_K}\boldsymbol{s}_K$.
return $\hat{\boldsymbol{x}}_K$ and $\tilde{\boldsymbol{y}}_K$.

and the primal-dual form

$$\min_{\boldsymbol{x} \geq \boldsymbol{0}_d} \max_{\boldsymbol{y} \in \mathbb{R}^n} \left\{\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{y}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}^T\boldsymbol{b}\right\}. \tag{PD-LP}$$

This problem has a sharpness property that can be used to obtain linear convergence in first-order methods [AHLL21]. For convenience, in the following, we define $\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{y}), \hat{\boldsymbol{w}} = (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}), \tilde{\boldsymbol{w}} = (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$ and $\boldsymbol{w}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$. Meanwhile, for $\gamma > 0$, we denote the weighted norm $\|\boldsymbol{w}\|_{(\gamma)} := \sqrt{\gamma\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 + \frac{1}{\gamma}\|\boldsymbol{y} - \boldsymbol{y}^*\|_2^2}$. Further, we use $\mathcal{W}^*$ to denote the optimal solution set of the LP and define the distance to $\mathcal{W}^*$ by $\mathrm{dist}(\boldsymbol{w}, \mathcal{W}^*)_{(\gamma)} = \min_{\boldsymbol{w}^* \in \mathcal{W}^*}\|\boldsymbol{w} - \boldsymbol{w}^*\|_{(\gamma)}$. When $\gamma = 1$, $\|\cdot\|_{(\gamma)}$ is

the standard Euclidean norm. Then based on (PD-LP), we can use the following classical LPMetric[4] to measure the progress of iterative algorithms for LP:

$$\begin{aligned}&\text{LPMetric}(\boldsymbol{x},\boldsymbol{y})\\&=\sqrt{\|\max\{-\boldsymbol{x},\boldsymbol{0}\}\|_2^2+\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2+\|\max\{-\boldsymbol{A}^T\boldsymbol{y}-\boldsymbol{c},\boldsymbol{0}\}\|_2^2+|\max\{\boldsymbol{c}^T\boldsymbol{x}+\boldsymbol{b}^T\boldsymbol{y},0\}|^2},\end{aligned}\tag{4.45}$$

which can be explicitly and directly computed. For the Euclidean case ($\gamma = 1$), it is well-known [Hof03] that there exists a Hoffman constant H_1 such that

$$\text{LPMetric}(\boldsymbol{w}) \geq H_1 \text{dist}(\boldsymbol{w}, \mathcal{W}^*)_{(1)}.\tag{4.46}$$

Using the equivalence of norms in finite dimensions, for general $\gamma > 0$, we can conclude that there exists another constant H_γ (to which we refer as the generalized Hoffman's constant) such that

$$\text{LPMetric}(\boldsymbol{w}) \geq H_\gamma \text{dist}(\boldsymbol{w}, \mathcal{W}^*)_{(\gamma)}.\tag{4.47}$$

Using Eq. (4.47) and Theorem 8, we then obtain the following bounds for distance and LPMetric.

The standard-form LP (PD-LP) is derived by setting $r(\boldsymbol{x}) \equiv 0$ and $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^d : x_i \geq 0, i \in [d]\}$ in (PD-GLP). Given any $\boldsymbol{w} \in \mathcal{X} \times \mathbb{R}^n$, we define a compact convex subset $\mathcal{W}_{\zeta,\gamma}(\boldsymbol{w})$ of $\mathcal{X} \times \mathbb{R}^n$ as follows:

$$\mathcal{W}_{\zeta,\gamma}(\boldsymbol{w}) = \{\hat{\boldsymbol{w}} \in \mathcal{X} \times \mathbb{R}^n : \|\hat{\boldsymbol{w}} - \boldsymbol{w}\|_{(\gamma)} \leq \zeta, \zeta > 0, \gamma > 0\},\tag{4.48}$$

where we have defined $\|\cdot\|_{(\gamma)}$ by $\|\boldsymbol{w}\|_{(\gamma)} = \sqrt{\gamma\|\boldsymbol{x}\|^2 + \frac{1}{\gamma}\|\boldsymbol{y}\|^2}$ in Section 4.4.3.

Lemma 33. *Consider the standard-form LP* (LP)*. Given $\tau > 0$ and $\boldsymbol{w} \in \mathcal{X} \times \mathbb{R}^n$, if $\|\boldsymbol{w}\|_{(\gamma)} \leq \tau$ and $\zeta \leq \tau$, then*

$$\sup_{\hat{\boldsymbol{w}} \in \mathcal{W}_{\zeta,\gamma}(\boldsymbol{w})} \{\mathcal{L}(\boldsymbol{x}, \hat{\boldsymbol{y}}) - \mathcal{L}(\hat{\boldsymbol{x}}, \boldsymbol{y})\} \geq \frac{\zeta}{\gamma + 1/\gamma + \tau} \text{LPMetric}(\boldsymbol{w}).\tag{4.49}$$

Proof. Let $\boldsymbol{F}(\boldsymbol{w}) = \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{c} \\ \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} \end{bmatrix}$. Then we have

$$\rho_{\zeta,\gamma}(\boldsymbol{w}) := \sup_{\hat{\boldsymbol{w}} \in \mathcal{W}_{\zeta,\gamma}(\boldsymbol{w})} \{\mathcal{L}(\boldsymbol{x}, \hat{\boldsymbol{y}}) - \mathcal{L}(\hat{\boldsymbol{x}}, \boldsymbol{y})\} = \sup_{\hat{\boldsymbol{w}} \in \mathcal{W}_{\zeta,\gamma}(\boldsymbol{w})} \boldsymbol{F}(\boldsymbol{w})^T(\boldsymbol{w} - \hat{\boldsymbol{w}}) \geq 0,\tag{4.50}$$

where the inequality follows from $\boldsymbol{w} \in \mathcal{W}_{\zeta,\gamma}(\boldsymbol{w})$.

[4] In (PD-LP), we dualize the constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ by $\boldsymbol{y}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$ instead of $\boldsymbol{y}^T(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})$, so in our LPMetric, there exist a sign difference for $\boldsymbol{y}$ from the more common representation such as the one in [AHLL21].

First, we prove

$$\rho_{\zeta,\gamma}(\boldsymbol{w}) \geq \frac{\zeta\|\max\{-\boldsymbol{A}^T\boldsymbol{y}-\boldsymbol{c},\,\boldsymbol{0}\}\|_2}{\gamma}. \tag{4.51}$$

If $\|\max\{-\boldsymbol{A}^T\boldsymbol{y}-\boldsymbol{c},\,\boldsymbol{0}\}\|_2 > 0$, for $\hat{\boldsymbol{w}}_1 = (\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{y}}_1)$, let $\hat{\boldsymbol{x}}_1 = \boldsymbol{x} + \frac{\zeta}{\gamma\|\max\{-\boldsymbol{A}^T\boldsymbol{y}-\boldsymbol{c},\,\boldsymbol{0}\}\|_2}\max\{-\boldsymbol{A}^T\boldsymbol{y}-\boldsymbol{c},\,\boldsymbol{0}\} \in \mathcal{X}$ and $\hat{\boldsymbol{y}}_1 = \boldsymbol{y}$. Then we have $\|\hat{\boldsymbol{w}}_1 - \boldsymbol{w}\|_{(\gamma)} = \zeta$ and thus $\hat{\boldsymbol{w}}_1 \in \mathcal{W}_{\zeta,\gamma}(\boldsymbol{w})$. So, as $\rho_{\zeta,\gamma}(\boldsymbol{w}) \geq \boldsymbol{F}(\boldsymbol{w})^T(\boldsymbol{w} - \hat{\boldsymbol{w}}_1)$, with the definition of $\hat{\boldsymbol{w}}_1$, Eq. (4.51) holds. If $\|\max\{-\boldsymbol{A}^T\boldsymbol{y} - \boldsymbol{c},\,\boldsymbol{0}\}\|_2 = 0$, then Eq. (4.51) holds trivially.

Second, we prove

$$\rho_{\zeta,\gamma}(\boldsymbol{w}) \geq \zeta\gamma\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2. \tag{4.52}$$

If $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 > 0$, for $\hat{\boldsymbol{w}}_2 = (\hat{\boldsymbol{x}}_2, \hat{\boldsymbol{y}}_2)$, let $\hat{\boldsymbol{x}}_2 = \boldsymbol{x}$ and $\hat{\boldsymbol{y}}_2 = \boldsymbol{y} + \frac{\zeta\gamma}{\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$. Then we have $\|\hat{\boldsymbol{w}}_2 - \boldsymbol{w}\|_{(\gamma)} = \zeta$ and thus $\hat{\boldsymbol{w}}_2 \in \mathcal{W}_{\zeta,\gamma}(\boldsymbol{w})$. Then as $\rho_{\zeta,\gamma}(\boldsymbol{w}) \geq \boldsymbol{F}(\boldsymbol{w})^T(\boldsymbol{w} - \hat{\boldsymbol{w}}_2)$, Eq. (4.52) holds. If $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 = 0$, then Eq. (4.52) holds trivially.

Third, we prove that

$$\rho_{\zeta,\gamma}(\boldsymbol{w}) \geq \frac{\zeta}{\tau}\max\{\boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{y},\,\boldsymbol{0}\}. \tag{4.53}$$

Note that the inequality holds trivially if $\boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{y} \leq 0$, so we assume that $\boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{y} > 0$, and note that $\boldsymbol{F}(\boldsymbol{w})^T\boldsymbol{w} = \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{y} > 0$ in this case. This condition implies that $\boldsymbol{w} \neq 0$, so we can define $\hat{\boldsymbol{w}}_3 = \boldsymbol{w} - \min\{\frac{\zeta}{\|\boldsymbol{w}\|_{(\gamma)}}, 1\}\boldsymbol{w}$. Then we have $\|\hat{\boldsymbol{w}}_3 - \boldsymbol{w}\|_{(\gamma)} \leq \frac{\zeta}{\|\boldsymbol{w}\|_{(\gamma)}}\|\boldsymbol{w}\|_{(\gamma)} \leq \zeta$. Meanwhile, we also have $\hat{\boldsymbol{x}}_3 \geq \boldsymbol{x}_3 - \boldsymbol{x}_3 = 0$, so that $\hat{\boldsymbol{w}}_3 \in \mathcal{X} \times \mathbb{R}^n$. Thus, we have $\hat{\boldsymbol{w}}_3 \in \mathcal{W}_{\zeta,\gamma}(\boldsymbol{w})$. Then, with the assumptions $\zeta \leq \tau$ and $\|\boldsymbol{w}\|_{(\gamma)} \leq \tau$, together with $\boldsymbol{F}(\boldsymbol{w})^T\boldsymbol{w} > 0$, we have

$$\rho_{\zeta,\gamma}(\boldsymbol{w}) \geq \min\Big\{\frac{\zeta}{\|\boldsymbol{w}\|_{(\gamma)}}, 1\Big\}\boldsymbol{F}(\boldsymbol{w})^T\boldsymbol{w} \geq \min\Big\{\frac{\zeta}{\tau}, 1\Big\}\boldsymbol{F}(\boldsymbol{w})^T\boldsymbol{w} = \frac{\zeta}{\tau}\boldsymbol{F}(\boldsymbol{w})^T\boldsymbol{w} = \frac{\zeta}{\tau}(\boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{y}), \tag{4.54}$$

completing the proof of the claim (4.53).

By combining Eqs. (4.51), (4.52) and (4.53), we obtain

$$\begin{aligned}(\gamma + 1/\gamma + \tau)^2\rho^2_{\zeta,\gamma}(\boldsymbol{w}) &\geq \zeta^2(\|\max\{-\boldsymbol{A}^T\boldsymbol{y} - \boldsymbol{c}, 0\}\|_2^2 + \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \max\{\boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{y}, 0\}^2) \\ &\geq \zeta^2(\text{LPMetric}(\boldsymbol{w}))^2,\end{aligned} \tag{4.55}$$

from which the result follows. □

Lemma 34. *Consider the standard-form LP* (LP)*, and let* $\boldsymbol{w}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ *be a Nash point (that is, a solution of the primal-dual form* (PD-LP)*). For a starting point* $\boldsymbol{w}_0$*, define* $\zeta = \|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_{(\gamma)}$ *and choose* $\gamma > 0$*. Then for all* $k = 1, 2, \ldots,$ *we have*

$$\mathbb{E}[\|\tilde{\boldsymbol{w}}_k - \boldsymbol{w}^*\|_{(\gamma)}] \leq \sqrt{2}\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_{(\gamma)},$$

where the expectation is taken w.r.t. all the randomness up to iteration k. Further, for $\mathcal{W}_{\zeta,\gamma}(\tilde{\boldsymbol{w}}_k)$ defined as in (4.48)*, we have*

$$\mathbb{E}\Big[\sup_{\boldsymbol{w}\in\mathcal{W}_{\zeta,\gamma}(\tilde{\boldsymbol{w}}_k)}\{\mathcal{L}(\tilde{\boldsymbol{x}}_k,\boldsymbol{y})-\mathcal{L}(\boldsymbol{y},\tilde{\boldsymbol{y}}_k)\}\Big]\leq\frac{25\hat{L}m}{k}\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}^2. \tag{4.56}$$

Proof. By Theorem 8, we have

$$\mathbb{E}\Big[\frac{\gamma}{4}\|\boldsymbol{x}^*-\boldsymbol{x}_k\|^2+\frac{1}{2\gamma}\|\boldsymbol{y}^*-\boldsymbol{y}_k\|^2\Big]\leq\frac{\gamma}{2}\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2+\frac{1}{2\gamma}\|\boldsymbol{y}^*-\boldsymbol{y}_0\|^2=\frac{1}{2}\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}^2,$$

by the definition of $\|\cdot\|_{(\gamma)}$. Using this definition again, we have that the left-hand side in this expression is bounded below by $\frac{1}{4}\|\boldsymbol{w}_k-\boldsymbol{w}^*\|_{(\gamma)}^2$, so that

$$\mathbb{E}\Big[\|\boldsymbol{w}_k-\boldsymbol{w}^*\|_{(\gamma)}^2]\leq 2\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}^2,\quad\forall k\geq 1. \tag{4.57}$$

By convexity of $\|\cdot\|^2$, we obtain

$$\mathbb{E}[\|\tilde{\boldsymbol{w}}_k-\boldsymbol{w}^*\|_{(\gamma)}^2]=\mathbb{E}\Big[\Big\|\frac{1}{k}\sum_{i=1}^k(\boldsymbol{w}_i-\boldsymbol{w}^*)\Big\|_{(\gamma)}^2\Big]\leq\frac{1}{k}\mathbb{E}\Big[\sum_{i=1}^k\|\boldsymbol{w}_i-\boldsymbol{w}^*\|_{(\gamma)}^2\Big]\leq 2\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}^2. \tag{4.58}$$

The first claim of the lemma now follows by applying Jensen's inequality to this bound.

For $\hat{\boldsymbol{w}}=(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})\in\mathcal{W}_{\zeta,\gamma}(\tilde{\boldsymbol{w}}_k)$, we have the following bound on $\mathbb{E}\Big[\gamma\|\hat{\boldsymbol{x}}-\boldsymbol{x}_0\|^2+\frac{1}{\gamma}\|\hat{\boldsymbol{y}}-\boldsymbol{y}_0\|^2\Big]$:

$$\begin{aligned}
&\mathbb{E}\Big[\gamma\|\hat{\boldsymbol{x}}-\boldsymbol{x}_0\|^2+\frac{1}{\gamma}\|\hat{\boldsymbol{y}}-\boldsymbol{y}_0\|^2\Big]\\
&=\mathbb{E}[\|\boldsymbol{w}_0-\hat{\boldsymbol{w}}\|_{(\gamma)}^2]\\
&=\mathbb{E}[\|\boldsymbol{w}_0-\boldsymbol{w}^*+\boldsymbol{w}^*-\tilde{\boldsymbol{w}}_k+\tilde{\boldsymbol{w}}_k-\hat{\boldsymbol{w}}\|_{(\gamma)}^2]\\
&\leq\mathbb{E}[(\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}+\|\boldsymbol{w}^*-\tilde{\boldsymbol{w}}_k\|_{(\gamma)}+\|\tilde{\boldsymbol{w}}_k-\hat{\boldsymbol{w}}\|_{(\gamma)})^2]\\
&\leq\mathbb{E}[3(\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}^2+\|\boldsymbol{w}^*-\tilde{\boldsymbol{w}}_k\|_{(\gamma)}^2+\|\tilde{\boldsymbol{w}}_k-\hat{\boldsymbol{w}}\|_{(\gamma)}^2)]\\
&\leq\mathbb{E}[3(\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}^2+2\|\boldsymbol{w}^*-\boldsymbol{w}_0\|_{(\gamma)}^2+\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}^2)]\\
&=12\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}^2,
\end{aligned} \tag{4.59}$$

where the first inequality is by the triangle inequality, the second inequality is by the arithmetic inequality, the third inequality is by Eq. (4.58) and our assumption $\hat{\boldsymbol{w}}\in\mathcal{W}_{\zeta,\gamma}(\tilde{\boldsymbol{w}}_k)$ with $\zeta=\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}$.

For the case of linear programming, the strong convexity constant is $\sigma=0$, so that $A_k=\frac{k}{2\hat{L}m}$ in CLVR. Thus, by applying Theorem 8 with $\mathcal{W}_k=\mathcal{W}_{\zeta,\gamma}(\tilde{\boldsymbol{w}}_k)$ and

$$\hat{\boldsymbol{w}}=(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})=\arg\sup_{\boldsymbol{w}=(\boldsymbol{x},\boldsymbol{y})\in\mathcal{W}_{\zeta,\gamma}(\tilde{\boldsymbol{w}}_k)}\{\mathcal{L}(\tilde{\boldsymbol{x}}_k,\boldsymbol{y})-\mathcal{L}(\boldsymbol{x},\tilde{\boldsymbol{y}}_k)\},$$

and using the definition of $\|\cdot\|_{(\gamma)}$ and Eq. (4.59), we have

$$\begin{aligned}
&\mathbb{E}\Big[\sup_{\boldsymbol{w}\in\mathcal{W}_{\zeta,\gamma}(\tilde{\boldsymbol{w}}_k)}\{\mathcal{L}(\tilde{\boldsymbol{x}}_k,\boldsymbol{y})-\mathcal{L}(\boldsymbol{y},\tilde{\boldsymbol{y}}_k)\}\Big]\\
&\le \frac{2\hat{L}m}{k}\Big(\mathbb{E}\Big[\frac{\gamma}{2}\|\hat{\boldsymbol{x}}-\boldsymbol{x}_0\|^2+\frac{1}{\gamma}\|\hat{\boldsymbol{y}}-\boldsymbol{y}_0\|^2\Big]+\frac{\gamma}{2}\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2+\frac{1}{2\gamma}\|\boldsymbol{y}^*-\boldsymbol{y}_0\|^2\Big)\\
&\le \frac{2\hat{L}m}{k}\Big(\mathbb{E}[\|\boldsymbol{w}_0-\hat{\boldsymbol{w}}\|_{(\gamma)}^2]+\frac{1}{2}\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}^2\Big)\\
&\le \frac{2\hat{L}m}{k}\Big(12\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}^2+\frac{1}{2}\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}^2\Big)\\
&\le \frac{25\hat{L}m}{k}\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}^2. \qquad (4.60)
\end{aligned}$$

□

Then with Theorem 8, Lemmas 33 and 34, we give our theorem for the fixed restart strategy.

Theorem 9. *Consider the* CLVR *algorithm applied to the standard-form LP problem* (PD-LP)*, with input* $\boldsymbol{w}_0$ *and output* $\tilde{\boldsymbol{w}}_k$*. Given* $\gamma>0$*, define* $\boldsymbol{w}^*=\arg\min_{w\in\mathcal{W}^*}\|\boldsymbol{w}_0-\boldsymbol{w}\|_{(\gamma)}$*, and define* $C_0=\gamma+1/\gamma+(\sqrt{2}+1)\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}+\|\boldsymbol{w}^*\|_{(\gamma)}$*. Then for* H_γ *defined as in* (4.47)*, we have*

$$\mathbb{E}\Big[\sqrt{\mathrm{dist}(\tilde{\boldsymbol{w}}_k,\mathcal{W}^*)_{(\gamma)}}\,\Big]\le 5\sqrt{\frac{\hat{L}mC_0}{H_\gamma k}}\sqrt{\mathrm{dist}(\boldsymbol{w}_0,\mathcal{W}^*)_{(\gamma)}},$$

$$\mathbb{E}[\sqrt{\mathrm{LPMetric}(\tilde{\boldsymbol{w}}_k)}]\le 5\sqrt{\frac{\hat{L}mC_0}{H_\gamma k}}\sqrt{\mathrm{LPMetric}(\boldsymbol{w}_0)}.$$

Proof. Applying Lemma 33 with $\boldsymbol{w}=\tilde{\boldsymbol{w}}_k$, $\tau=\|\tilde{\boldsymbol{w}}_k\|_{(\gamma)}+\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}$ and $\zeta=\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}\le\tau$, we have

$$\sup_{\hat{\boldsymbol{w}}\in\mathcal{W}_{\zeta,\gamma}(\boldsymbol{w})}\{\mathcal{L}(\tilde{\boldsymbol{x}}_k,\hat{\boldsymbol{y}})-\mathcal{L}(\hat{\boldsymbol{x}},\tilde{\boldsymbol{y}}_k)\}\ge\frac{\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}\mathrm{LPMetric}(\tilde{\boldsymbol{w}}_k)}{\gamma+1/\gamma+\|\tilde{\boldsymbol{w}}_k\|_{(\gamma)}+\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}}. \qquad (4.61)$$

As $\frac{x^2}{y}$ is jointly convex in x,y on the domain $\{(x,y):x\in\mathbb{R},y>0\}$ [BV04, Example 3.18] using Jensen's inequality, we have that $\mathbb{E}[\frac{x^2}{y}]\ge\frac{(\mathbb{E}[x])^2}{\mathbb{E}[y]}$. (In our case, this simply follows as x,y depend on the same source of randomness.) Applying this inequality to (4.61) with $x=\sqrt{\mathrm{LPMetric}(\tilde{\boldsymbol{w}}_k)}$ and $y=\gamma+1/\gamma+\|\tilde{\boldsymbol{w}}_k\|_{(\gamma)}+\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}$, we obtain

$$\begin{aligned}
\mathbb{E}\Big[\sup_{\hat{\boldsymbol{w}}\in\mathcal{W}_{\zeta,\gamma}(\tilde{\boldsymbol{w}}_k)}\{\mathcal{L}(\tilde{\boldsymbol{x}}_k,\hat{\boldsymbol{y}})-\mathcal{L}(\hat{\boldsymbol{x}},\tilde{\boldsymbol{y}}_k)\}\Big]&\ge\mathbb{E}\Big[\frac{\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}(\sqrt{\mathrm{LPMetric}(\tilde{\boldsymbol{w}}_k)})^2}{\gamma+1/\gamma+\|\tilde{\boldsymbol{w}}_k\|_{(\gamma)}+\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}}\Big]\\
&\ge\frac{\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}\Big(\mathbb{E}[\sqrt{\mathrm{LPMetric}(\tilde{\boldsymbol{w}}_k)}]\Big)^2}{\mathbb{E}[\gamma+1/\gamma+\|\tilde{\boldsymbol{w}}_k\|_{(\gamma)}+\|\boldsymbol{w}_0-\boldsymbol{w}^*\|_{(\gamma)}]}. \qquad (4.62)
\end{aligned}$$

Using Lemma 34, we have

$$\begin{aligned}
&\mathbb{E}[\gamma + 1/\gamma + \|\tilde{\boldsymbol{w}}_k\|_{(\gamma)} + \|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_{(\gamma)}] \\
&= \gamma + 1/\gamma + \|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_{(\gamma)} + \mathbb{E}[\|\tilde{\boldsymbol{w}}_k\|_{(\gamma)}] \\
&\leq \gamma + 1/\gamma + \|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_{(\gamma)} + \mathbb{E}[\|\boldsymbol{w}^*\|_{(\gamma)} + \|\tilde{\boldsymbol{w}}_k - \boldsymbol{w}^*\|_{(\gamma)}] \\
&\leq \gamma + 1/\gamma + \|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_{(\gamma)} + \|\boldsymbol{w}^*\|_{(\gamma)} + \sqrt{2}\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_{(\gamma)} \\
&= \gamma + 1/\gamma + (\sqrt{2}+1)\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_{(\gamma)} + \|\boldsymbol{w}^*\|_{(\gamma)}. \qquad (4.63)
\end{aligned}$$

By combining Eqs. (4.62) and (4.63) and using the definition of C_0, we have

$$\begin{aligned}
\mathbb{E}\Big[\sup_{\hat{\boldsymbol{w}} \in \mathcal{W}_{\zeta,\gamma}(\tilde{\boldsymbol{w}}_k)} \{\mathcal{L}(\tilde{\boldsymbol{x}}_k, \hat{\boldsymbol{y}}) - \mathcal{L}(\hat{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_k)\}\Big] &\geq \frac{\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_{(\gamma)}}{C_0} \Big(\mathbb{E}[\sqrt{\mathrm{LPMetric}(\tilde{\boldsymbol{w}}_k)}]\Big)^2 \\
&\geq \frac{\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_{(\gamma)}}{C_0} \Big(\mathbb{E}[\sqrt{H_\gamma \mathrm{dist}(\tilde{\boldsymbol{w}}_k, \mathcal{W}^*)_{(\gamma)}}]\Big)^2, \qquad (4.64)
\end{aligned}$$

where the last inequality is by the definition of Hoffman constant in Eq. (4.47). Meanwhile, by Lemma 34, we have:

$$\mathbb{E}\Big[\sup_{\hat{\boldsymbol{w}} \in \mathcal{W}_{\zeta,\gamma}(\tilde{\boldsymbol{w}}_k)} \{\mathcal{L}(\tilde{\boldsymbol{x}}_k, \hat{\boldsymbol{y}}) - \mathcal{L}(\hat{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_k)\}\Big] \leq \frac{25\hat{L}m}{k} \|\boldsymbol{w}_0 - \boldsymbol{w}^*\|^2_{(\gamma)}. \qquad (4.65)$$

Now, recalling that $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w} \in \mathcal{W}^*} \|\boldsymbol{w}_0 - \boldsymbol{w}\|_{(\gamma)}$ and using Eq. (4.47), we have

$$\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_{(\gamma)} = \mathrm{dist}(\boldsymbol{w}_0, \mathcal{W}^*)_{(\gamma)} \leq \frac{1}{H_\gamma} \mathrm{LPMetric}(\boldsymbol{w}_0). \qquad (4.66)$$

By combining Eqs. (4.64)-(4.66), we have

$$\begin{aligned}
\frac{1}{C_0}\Big(\mathbb{E}\Big[\sqrt{H_\gamma \mathrm{dist}(\tilde{\boldsymbol{w}}_k, \mathcal{W}^*)_{(\gamma)}}\Big]\Big)^2 &\leq \frac{1}{C_0}\Big(\mathbb{E}[\sqrt{\mathrm{LPMetric}(\tilde{\boldsymbol{w}}_k)}]\Big)^2 \\
&\leq \frac{25\hat{L}m}{k} \mathrm{dist}(\boldsymbol{w}_0, \mathcal{W}^*)_{(\gamma)} \\
&\leq \frac{25\hat{L}m}{H_\gamma k} \mathrm{LPMetric}(\boldsymbol{w}_0). \qquad (4.67)
\end{aligned}$$

Both bounds follow from this chain of inequalities. □

As a result, by Theorem 9, if we know the values of $\hat{L}, \|\boldsymbol{w}^*\|_{(\gamma)}$ and H_γ, then by setting $k = \frac{100\hat{L}mC_0}{H_\gamma k}$, we can halve the square root of the distance and the LPMetric in expectation. Thus we can obtain linear convergence if we restart the CLVR algorithm after a fixed number of iterations. However, the values of $\|\boldsymbol{w}^*\|_{(\gamma)}$ and H_γ are often unknown and thus make this strategy unrealistic in practice.

Compared with the above fixed restart strategy, a natural strategy is to restart whenever the LPMetric halves, summarized in Algorithm 11 below. Since LPMetric is easy to monitor and update, implementation of this strategy is straightforward. However, bounding the number of iterations required to halve the metric (in expectation or with high probability) seems nontrivial. What can be said (based on Theorem 9 and denoting by K the number of iterations on CLVR between restarts) is that $\mathbb{P}[K > \frac{50\hat{L}mC_0}{\delta^2 H_\gamma}] \leq \delta$. This follows by Markov inequality, as $\mathbb{P}[K > k] = \mathbb{P}\Big[\sqrt{\text{LPMetric}(\tilde{\boldsymbol{w}}_k)} > \sqrt{\frac{\text{LPMetric}(\boldsymbol{w}_0)}{2}}\Big] \leq 5\sqrt{2\frac{\hat{L}mC_0}{H_\gamma k}}$. We provide a comparison between the adaptive restart scheme proposed in [AHLL21] and our proposed adaptive restart scheme in Section 4.6.1 to demonstrate its practical competitiveness. Although we use adaptive restart in our experiments, we defer its convergence analysis to future work. Finally, as an independent and parallel work to ours, [LY21] proposed a high probability guarantee for scheduled restart for stochastic extragradient-type methods.

Finally, we summarize the adaptive restart strategy in Algorithm 11.

Algorithm 11 Lazy CLVR with Adaptive Restarts

Input: $\epsilon > 0$, $\boldsymbol{x}_0 \in \mathcal{X}, \boldsymbol{y}_0 \in \mathbb{R}^n$, $\gamma > 0$, $\hat{L} > 0$, K, $\widehat{K}$, m, $\{S^1, S^2, \ldots, S^m\}$, $\{C^1, C^2, \ldots, C^m\}$
$t = 0$, $\boldsymbol{x}_0^{(0)} = \boldsymbol{x}_0$, $\boldsymbol{y}_0^{(0)} = \boldsymbol{y}_0$, $\boldsymbol{w}^{(0)} = (\boldsymbol{x}_0^{(0)}, \boldsymbol{y}_0^{(0)})$
repeat
    Run Lazy CLVR (Algorithm 10) until $\text{LPMetric}(\boldsymbol{w}^{(t+1)}) \leq \frac{1}{2}\text{LPMetric}(\boldsymbol{w}^{(t)})$ where, $\boldsymbol{w}^{(t+1)} = \tilde{\boldsymbol{w}}_K^{(t)} = (\tilde{\boldsymbol{x}}_K^{(t)}, \tilde{\boldsymbol{y}}_K^{(t)})$ and $\tilde{\boldsymbol{x}}_K^{(t)}, \tilde{\boldsymbol{y}}_K^{(t)}$ are the output points of Lazy CLVR
    $t = t + 1$
until $\text{LPMetric}(\boldsymbol{w}^{(t)}) \leq \epsilon$
Return: $\boldsymbol{w}^{(t)}$

4.5 Application: Distributionally Robust Optimization

The underlying assumption of standard empirical minimization in machine learning is that all the data samples (used for both training and testing) are sampled uniformly at random from a *fixed* probability distribution [Vap13]. However, it is well-known that such an assumption is often violated in practice—the underlying distribution is not fixed and can vary due to a variety of reasons, such as changing from the underlying domain [BDBC$^+$10], perturbations from the physical world [GSS14, KGB$^+$16, MMS$^+$18], and adversarial attacks [KGB17, CW17]. To address the uncertainty of the underlying distribution, a more reasonable assumption is that the distribution itself can vary within some ambiguity set while the goal for the trained model is that it performs well even w.r.t. the worst case distribution from the ambiguity set. Such a learning paradigm is called *Distributionally Robust Optimization (DRO)* [DY10, SAMEK15, ND16, DN21, HNSS18,

EK18, SJ19, DN19, LCDS20] and the resulting optimization problem based on finite data is called *Robust Empirical Risk Minimization (R-ERM)* [ND16]. The ambiguity set is typically modeled by bounded deviation from the starting (e.g., uniform) distribution, where deviation is measured using functions such as the f-divergence [ND16, HNSS18, DN21, LCDS20] and Wasserstein metric [SAMEK15, EK18, LHS19, YLMJ21]. In this paper, we focus on the ambiguity sets defined w.r.t. the Wasserstein metric, as this setup has gained significant attention in both theory [PC$^+$19] and practice [ACB17], as it allows the two distributions to be defined on different support sets.

Despite its substantial success in operations research [DY10, WKS14, BGK18, EK18, DGN21], DRO has not been widely adopted in common machine learning practice due to the lack of large scale optimization methods for solving the corresponding R-ERM problem.

Consider sample vectors $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\}$ with labels $\{b_1, b_2, \ldots, b_n\}$, where $b_i \in \{1, -1\}\,(i \in [n])$. The DRO problem with f-divergence based ambiguity set is

$$\min_{\boldsymbol{x}\in\mathcal{X}} \sup_{\boldsymbol{p}\in\mathcal{P}_{\rho,n}} \sum_{i=1}^{n} p_i g(b_i \boldsymbol{a}_i^T \boldsymbol{x}), \tag{4.68}$$

where $\mathcal{P}_{\rho,n} = \{\boldsymbol{p} \in \mathbb{R}^n : \sum_{i=1}^n p_i = 1, p_i \geq 0\,(i \in [n]), D_f(\boldsymbol{p}\|\mathbf{1}/n) \leq \frac{\rho}{n}\}$ is the ambiguity set, g is a convex loss function and D_f is an f-divergence defined by $D_f(\boldsymbol{p}\|\boldsymbol{q}) = \sum_{i=1} q_i f(p_i/q_i)$ with $\boldsymbol{p}, \boldsymbol{q} \in \{\boldsymbol{p} \in \mathbb{R}^n : \sum_{i=1}^n p_i = 1, p_i \geq 0\}$ and f being a convex function [ND16]. The formulation (4.68) is a nonbilinearly coupled convex-concave min-max problem with constraint set $\mathcal{P}_{\rho,n}$ for which efficient projections are not available in general. When g is a nonsmooth loss (e.g., the hinge loss), many well-known methods such as the extragradient [Kor76, Nem04] cannot be used even if we could project onto $\mathcal{P}_{\rho,n}$ efficiently. However, by introducing auxiliary variables, additional linear constraints, and simple convex constraints, we can make the interacting term between primal and dual variables bilinear, as shown next.

Theorem 10. *Let $\mathcal{X}$ be a compact convex set. Then the DRO problem in Eq.* (4.68) *is equivalent to*

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{v},\boldsymbol{w},\boldsymbol{\mu},\boldsymbol{q},\gamma} \quad & \Big\{\gamma + \frac{\rho\mu_1}{n} + \frac{1}{n}\sum_{i=1}^{n} \mu_i f^*\Big(\frac{q_i}{\mu_i}\Big)\Big\} \\
\text{s.t.} \quad & \boldsymbol{w} + \boldsymbol{v} - \frac{\boldsymbol{q}}{n} - \gamma\mathbf{1}_n = \mathbf{0}_n, \\
& u_i = b_i \boldsymbol{a}_i^T \boldsymbol{x}, && i \in [n] \\
& \mu_1 = \mu_2 = \cdots = \mu_n, \\
& g(u_i) \leq w_i, && i \in [n] \\
& q_i \in \mu_i \mathrm{dom}(f^*), && i \in [n] \\
& v_i \geq 0,\ \mu_i \geq 0, && i \in [n] \\
& \boldsymbol{x} \in \mathcal{X}.
\end{aligned}$$

Proof. Since the context is clear, to simplify the notation, in the following we use $\mathcal{P}$ to denote $\mathcal{P}_{\rho,n}$. First, using Sion's minimax theorem, we have that

$$\min_{\boldsymbol{x}\in\mathcal{X}}\sup_{\boldsymbol{p}\in\mathcal{P}}\sum_{i=1}^{n}p_i g(b_i\boldsymbol{a}_i^T\boldsymbol{x}) = \sup_{\boldsymbol{p}\in\mathcal{P}}\min_{\boldsymbol{x}\in\mathcal{X}}\sum_{i=1}^{n}p_i g(b_i\boldsymbol{a}_i^T\boldsymbol{x}).$$

Introducing auxiliary variables $\boldsymbol{w}$ and $\boldsymbol{u}$, the problem is further equivalent to

$$\sup_{\boldsymbol{p}\in\mathcal{P}}\ \min_{\substack{\boldsymbol{x}\in\mathcal{X},\boldsymbol{w},\boldsymbol{u}:\\ u_i=b_i\boldsymbol{a}_i^T\boldsymbol{x},i\in[n],\\ g(u_i)\le w_i,i\in[n]}}\ \boldsymbol{p}^T\boldsymbol{w}\ \equiv\ \min_{\substack{\boldsymbol{x}\in\mathcal{X},\boldsymbol{w},\boldsymbol{u}:\\ u_i=b_i\boldsymbol{a}_i^T\boldsymbol{x},i\in[n],\\ g(u_i)\le w_i,i\in[n]}}\ \sup_{\boldsymbol{p}\in\mathcal{P}}\ \boldsymbol{p}^T\boldsymbol{w},$$

where the last equivalence is by applying minimax equality, which holds due to compactness of $\mathcal{P}$ [Sto63]. Hence, we can conclude that

$$\min_{\boldsymbol{x}\in\mathcal{X}}\sup_{\boldsymbol{p}\in\mathcal{P}}\sum_{i=1}^{n}p_i g(b_i\boldsymbol{a}_i^T\boldsymbol{x})\ =\ \min_{\substack{\boldsymbol{x}\in\mathcal{X},\boldsymbol{w},\boldsymbol{u}:\\ u_i=b_i\boldsymbol{a}_i^T\boldsymbol{x},i\in[n],\\ g(u_i)\le w_i,i\in[n]}}\ \sup_{\boldsymbol{p}\in\mathcal{P}}\ \boldsymbol{p}^T\boldsymbol{w}. \tag{4.69}$$

For a fixed tuple $(\boldsymbol{x},\boldsymbol{w},\boldsymbol{u})$, using Lagrange multipliers to enforce the constraints from $\mathcal{P}$, we can further write

$$\begin{aligned}
&\sup_{\boldsymbol{p}\in\mathcal{P}}\boldsymbol{p}^T\boldsymbol{w} = -\inf_{\boldsymbol{p}\in\mathcal{P}}-\boldsymbol{p}^T\boldsymbol{w}\\
=\ &-\inf_{\boldsymbol{p}\in\mathbb{R}^n}\Big(-\boldsymbol{p}^T\boldsymbol{w}+\sup_{\boldsymbol{v}\ge 0,\gamma\in\mathbb{R},\mu\ge 0}\Big(-\boldsymbol{p}^T\boldsymbol{v}+\gamma\Big(\sum_{i=1}^{n}p_i-1\Big)+\mu\Big(D_f(\boldsymbol{p}\|\mathbf{1}/n)-\frac{\rho}{n}\Big)\Big)\Big)\\
=\ &\sup_{\boldsymbol{p}\in\mathbb{R}^n}\inf_{\boldsymbol{v}\ge 0,\gamma\in\mathbb{R},\mu\ge 0}\Big(\boldsymbol{p}^T\boldsymbol{w}+\boldsymbol{p}^T\boldsymbol{v}-\gamma\Big(\sum_{i=1}^{n}p_i-1\Big)-\mu\Big(D_f(\boldsymbol{p}\|\mathbf{1}/n)-\frac{\rho}{n}\Big)\Big).
\end{aligned}$$

Now, using the definitions of D_f and the convex conjugate of f, we have

$$D_f(\boldsymbol{p}\|\mathbf{1}/n)=\sum_{i=1}^{n}\frac{1}{n}f(np_i)=\sum_{i=1}^{n}\frac{1}{n}\sup_{\nu_i\in\mathrm{dom}(f^*)}(np_i\nu_i-f^*(\nu_i)). \tag{4.70}$$

As a result, we have

$$\begin{aligned}
&\inf_{\mu\ge 0}\ -\mu\Big(D_f(\boldsymbol{p}\|\mathbf{1}/n)-\frac{\rho}{n}\Big)\\
=\ &\inf_{\mu\ge 0}-\frac{\mu}{n}\Big(\sum_{i=1}^{n}\sup_{\nu_i\in\mathrm{dom}(f^*),i\in[n]}(np_i\nu_i-f^*(\nu_i))-\rho\Big)\\
=\ &\inf_{\mu\ge 0,\nu_i\in\mathrm{dom}(f^*),i\in[n]}\ -\frac{\mu}{n}\Big(\sum_{i=1}^{n}(np_i\nu_i-f^*(\nu_i))-\rho\Big)\\
=\ &\inf_{\mu\ge 0,q_i\in\mu\mathrm{dom}(f^*),i\in[n]}\ \frac{1}{n}\sum_{i=1}^{n}(-p_iq_i+\mu f^*(\frac{q_i}{n\mu}))+\frac{\mu\rho}{n}\\
=\ &\inf_{\substack{\mu_1=\mu_2=\cdots=\mu_n\ge 0,\\ q_i\in\mu_i\mathrm{dom}(f^*),i\in[n]}}\ \frac{1}{n}\sum_{i=1}^{n}(-p_iq_i+\mu_i f^*(\frac{q_i}{n\mu_i}))+\frac{\mu_1\rho}{n},
\end{aligned}$$

where the first equality is by Eq. (4.70), the third one is by the variable substitution $q_i = n\mu\nu_i$, and the last one is by introducing $\mu_1, \mu_2, \ldots, \mu_n$ to replace μ. Then each $n\mu_i f^*(\frac{q_i}{n\mu_i})(i \in [n])$ is a perspective function of f^* [BV04], which is jointly convex w.r.t. (μ_i, q_i). Hence, we can conclude that

$$
\begin{aligned}
& \sup_{\boldsymbol{p}\in\mathcal{P}} \boldsymbol{p}^T\boldsymbol{w} \\
= & \sup_{\boldsymbol{p}\in\mathbb{R}^n} \inf_{\substack{\gamma\in\mathbb{R},\boldsymbol{v}\geq\boldsymbol{0},\\ \mu_1=\mu_2=\cdots=\mu_n\geq 0,\\ q_i\in\mu_i\mathrm{dom}(f^*),i\in[n]}} \Big(\boldsymbol{p}^T\boldsymbol{w} + \boldsymbol{p}^T\boldsymbol{v} - \gamma\Big(\sum_{i=1}^n p_i - 1\Big) + \frac{1}{n}\sum_{i=1}^n (-p_i q_i + \mu_i f^*(\frac{q_i}{n\mu_i})) + \frac{\mu_1\rho}{n}\Big) \\
= & \inf_{\substack{\gamma\in\mathbb{R},\boldsymbol{v}\geq\boldsymbol{0},\\ \mu_1=\mu_2=\cdots=\mu_n\geq 0,\\ q_i\in\mu_i\mathrm{dom}(f^*),i\in[n]}} \sup_{\boldsymbol{p}\in\mathbb{R}^n} \Big(\boldsymbol{p}^T\boldsymbol{w} + \boldsymbol{p}^T\boldsymbol{v} - \gamma\Big(\sum_{i=1}^n p_i - 1\Big) + \frac{1}{n}\sum_{i=1}^n (-p_i q_i + \mu_i f^*(\frac{q_i}{n\mu_i})) + \frac{\mu_1\rho}{n}\Big),
\end{aligned}
$$

where the last line is by strong duality. Thus, combining with Eq. (4.69), we conclude that the original DRO problem with f-divergence based ambiguity set is equivalent to the following problem:

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{v},\boldsymbol{w},\boldsymbol{\mu},\boldsymbol{q},\gamma} \max_{\boldsymbol{p}\in\mathbb{R}^n} \ & \Big\{\gamma + \frac{\rho\mu_1}{n} + \frac{1}{n}\sum_{i=1}^n \mu_i f^*\Big(\frac{q_i}{\mu_i}\Big) + \boldsymbol{p}^T\Big(\boldsymbol{w} + \boldsymbol{v} - \frac{\boldsymbol{q}}{n} - \gamma\boldsymbol{1}_n\Big)\Big\} \\
\text{s.t.} \ & u_i = b_i\boldsymbol{a}_i^T\boldsymbol{x}, && i \in [n] \\
& \mu_1 = \mu_2 = \cdots = \mu_n, \\
& g(u_i) \leq w_i, && i \in [n] \\
& q_i \in \mu_i\mathrm{dom}(f^*), && i \in [n] \\
& v_i \geq 0, \mu_i \geq 0, && i \in [n] \\
& \boldsymbol{x} \in \mathcal{X}.
\end{aligned}
$$

Finally, noticing that the maximization problem over $\boldsymbol{p} \in \mathbb{R}^n$ enforces the equality constraint $\boldsymbol{w} + \boldsymbol{v} - \frac{\boldsymbol{q}}{n} - \gamma\boldsymbol{1}_n = \boldsymbol{0}_n$, we obtain

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{v},\boldsymbol{\mu},\boldsymbol{q},\gamma} \ & \Big\{\gamma + \frac{\rho\mu_1}{n} + \frac{1}{n}\sum_{i=1}^n \mu_i f^*(\frac{q_i}{\mu_i})\Big\} \\
\text{s.t.} \ & \boldsymbol{w} + \boldsymbol{v} - \frac{\boldsymbol{q}}{n} - \gamma\boldsymbol{1}_n = \boldsymbol{0}_n, \\
& u_i = b_i\boldsymbol{a}_i^T\boldsymbol{x}, && i \in [n] \\
& \mu_1 = \mu_2 = \cdots = \mu_n, \\
& g(u_i) \leq w_i, && i \in [n] \\
& q_i \in \mu_i\mathrm{dom}(f^*), && i \in [n] \\
& v_i \geq 0, \mu_i \geq 0, && i \in [n] \\
& \boldsymbol{x} \in \mathcal{X},
\end{aligned}
$$

as claimed. □

In Theorem 10, the domain of the one-dimensional convex function $f^*(\cdot)$ is an interval such as $[a, b]$, so that $q_i \in \mu_i \mathrm{dom}(f^*)$ denotes the inequality $\mu_i a \le q_i \le \mu_i b$. Since the perspective function $\mu f^*(\frac{q}{\mu})$ is a simple convex function of two variables, we can assume that the proximal operator for this function on the domain $\{(\mu, q) : q \in \mu \mathrm{dom}(f^*), \mu > 0\}$ can be computed efficiently [BV04]. Similarly, we can assume that the constraint $g(u) \le w$ admits an efficiently computable projection operator. As a result, the formulation (4.68) can be solved by CLVR. When expressing (4.68) in the form of (PD-GLP), the primal and dual variable vectors have dimensions $d + 1 + 4n$ and $3n - 1$, respectively. However, according to Table 4.1, provided that $\mathcal{X}$ is coordinate separable, the overall complexity of CLVR will only be $O(\frac{(\mathrm{nnz}(\boldsymbol{A})+n)(R+1)}{\epsilon})$.

Example: Conditional Value at Risk (CVaR) with hinge loss. As a specific example of an application of Theorem 10, we consider CVaR at level $\alpha \in (0, 1)$, which leads to the optimization problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \sup_{\substack{\boldsymbol{p}\ge 0, \mathbf{1}_n^T\boldsymbol{p}=1 \\ p_i \le \frac{1}{\alpha n}(i\in[d])}} \sum_{i=1}^n p_i g(b_i \boldsymbol{a}_i^T \boldsymbol{x}), \tag{4.71}$$

where $g(u_i) = \max\{0, 1 - u_i\}$ is the hinge loss. Here the ambiguity set constraint reduces to simple bounds $p_i \le \frac{1}{\alpha n}$ for $i \in [n]$, so the reformulation based on the convex perspective function can be avoided altogether and replaced by simple Lagrange multipliers for this linear constraint. In particular, in the proof of Theorem 10, we can write

$$\begin{aligned}
\sup_{\boldsymbol{p}\in\mathcal{P}} \boldsymbol{p}^T\boldsymbol{w} &= \sup_{\boldsymbol{p}\in\mathbb{R}^n} \inf_{\boldsymbol{v}\ge 0, \gamma\in\mathbb{R}, \boldsymbol{\mu}\ge 0} \Big(\boldsymbol{p}^T\boldsymbol{w} + \boldsymbol{p}^T\boldsymbol{v} - \gamma\Big(\sum_{i=1}^n p_i - 1\Big) - \sum_{i=1}^n \mu_i\Big(p_i - \frac{1}{\alpha n}\Big)\Big) \\
&= \sup_{\boldsymbol{p}\in\mathbb{R}^n} \inf_{\boldsymbol{v}\ge 0, \gamma\in\mathbb{R}, \boldsymbol{\mu}\ge 0} \Big(\gamma + \boldsymbol{p}^T(\boldsymbol{w} + \boldsymbol{v} - \gamma\mathbf{1}_n - \boldsymbol{\mu}) + \frac{1}{\alpha n}\boldsymbol{\mu}^T\mathbf{1}_n\Big)\Big) \\
&= \inf_{\boldsymbol{v}\ge 0, \gamma\in\mathbb{R}, \boldsymbol{\mu}\ge 0} \sup_{\boldsymbol{p}\in\mathbb{R}^n} \Big(\gamma + \boldsymbol{p}^T(\boldsymbol{w} + \boldsymbol{v} - \gamma\mathbf{1}_n - \boldsymbol{\mu}) + \frac{1}{\alpha n}\boldsymbol{\mu}^T\mathbf{1}_n\Big)\Big).
\end{aligned}$$

Following the argument from the proof of Theorem 10, and expressing $w_i \ge g(u_i)$ equivalently as the pair of constraints $w_i \ge 0$ and $w_i \ge 1 - u_i$, the problem reduces to

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{v},\boldsymbol{w},\boldsymbol{\mu},\gamma} \ & \Big\{\gamma + \frac{1}{\alpha n}\boldsymbol{\mu}^T\mathbf{1}_n\Big\} \\
\text{s.t.} \ & \boldsymbol{w} + \boldsymbol{v} - \gamma\mathbf{1}_n - \boldsymbol{\mu} = \mathbf{0}_n, \\
& u_i = b_i \boldsymbol{a}_i^T \boldsymbol{x}, && i \in [n], \\
& w_i \ge 0, \ w_i \ge 1 - u_i, && i \in [n], \\
& v_i \ge 0, \ \mu_i \ge 0, && i \in [n],
\end{aligned} \tag{4.72}$$

which is a linear program. To write it in the standard form, we further introduce slack variables $\boldsymbol{s} \in \mathbb{R}^n$, $\boldsymbol{s} \ge \mathbf{0}$, to replace inequality constraints $w_i \ge 1 - u_i$ by $s_i - u_i - w_i = -1$. For implementation

purposes, we define $\mathcal{X}$ to be the set of simple non-negativity constraints ($w_i \geq 0, s_i \geq 0,, v_i \geq 0, \mu_i, \geq 0$, $\forall i \in [n]$) from Eq. (4.72). The problem then becomes

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{v},\boldsymbol{w},\boldsymbol{\mu},\boldsymbol{s},\gamma} \quad & \Big\{\gamma + \frac{1}{\alpha n}\boldsymbol{\mu}^T \mathbf{1}_n\Big\} \\
\text{s. t.} \quad & \boldsymbol{w} + \boldsymbol{v} - \gamma \mathbf{1}_n - \boldsymbol{\mu} = \mathbf{0}_n, \\
& u_i - b_i \boldsymbol{a}_i^T \boldsymbol{x} = 0, \qquad\qquad i \in [n], \\
& \boldsymbol{s} - \boldsymbol{u} - \boldsymbol{w} = -\mathbf{1}_n, \\
& \boldsymbol{w} \geq \mathbf{0}_n,\ \boldsymbol{v} \geq \mathbf{0}_n,\ \boldsymbol{\mu} \geq \mathbf{0}_n,\ \boldsymbol{s} \geq \mathbf{0}_n.
\end{aligned}$$

The original DRO problem with Wasserstein metric based ambiguity set is an *infinite*-dimensional *nonbilinearly* coupled convex-concave min-max problem defined by

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \sup_{\mathbb{P}\in\mathcal{P}_{\rho,\kappa}} \mathbb{E}^{\mathbb{P}}[g(b\boldsymbol{a}^T\boldsymbol{w})], \tag{4.73}$$

where $\boldsymbol{a} \in \mathbb{R}^d, b \in \{1, -1\}$, $\mathbb{P}$ is a distribution on $\mathbb{R}^d \times \{1, -1\}$, g is a convex loss function and $\mathcal{P}_{\rho,\kappa}$ is the Wasserstein metric-based ambiguity set [SAMEK15].

Definition 7. *Let $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ be two probability distributions supported on $\Theta = \mathbb{R}^d \times \{1, -1\}$ and let $\Pi(\boldsymbol{\mu}, \boldsymbol{\nu})$ denote the set of all joint distributions between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$. Then the Wasserstein metric between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ is defined by*

$$W(\boldsymbol{\mu}, \boldsymbol{\nu}) = \inf_{\pi\in\Pi(\boldsymbol{\mu},\boldsymbol{\nu})} \int_{\Theta\times\Theta} \zeta(\boldsymbol{\xi}, \boldsymbol{\xi}')\pi(d\boldsymbol{\xi}, d\boldsymbol{\xi}'), \tag{4.74}$$

where $\boldsymbol{\xi}, \boldsymbol{\xi}' \in \Theta$ and $\zeta(\cdot, \cdot) : \Theta \times \Theta \to \mathbb{R}_+$ is a convex cost function defined by

$$\zeta((\boldsymbol{a}, b), (\boldsymbol{a}', b')) = \|\boldsymbol{a} - \boldsymbol{a}'\| + \kappa|b - b'|,$$

where $\|\cdot\|$ denotes a general norm and $\kappa > 0$ is used to balance the feature mismatch and label uncertainty. Let $\mathbb{Q} = \frac{1}{n}\sum_{i=1}^n \delta_{(\boldsymbol{a}_i, b_i)}$, where $\delta_{(\boldsymbol{a}_i,b_i)}$ is the Dirac Delta function (or a point mass) at point $(\boldsymbol{a}_i, b_i)$. Then, the Wasserstein metric based ambiguity set is defined as

$$\mathcal{P}_{\rho,\kappa} = \Big\{\mathbb{P} : W(\mathbb{P}, \mathbb{Q}) \leq \rho\Big\}. \tag{4.75}$$

Assumption 13. $\mathbb{R} \ni M = \sup_{\theta\in\mathrm{dom}(g^*)}|\theta|$.

We first show the following auxiliary lemma which is used in the proof of Theorem 11. A similar result for the special case of a logistic loss function can be found in [SAMEK15].

Lemma 35. *Let $(\boldsymbol{a}', b') \in \mathbb{R}^d \times \{1, -1\}$ be a given pair of data sample and label. Then, for every $\lambda > 0$, we have*

$$\sup_{\boldsymbol{a}\in\mathbb{R}^d} g(b'\boldsymbol{a}^T\boldsymbol{w}) - \lambda\|\boldsymbol{a}-\boldsymbol{a}'\| = \begin{cases} g(b'\boldsymbol{a}'^T\boldsymbol{w}), & \text{if } \|\boldsymbol{w}\|_* \leq \lambda/M \\ +\infty, & \text{otherwise} \end{cases}. \tag{4.76}$$

Proof. Since g is assumed to be proper, convex, and lower semicontinuous, by the Fenchel-Moreau theorem, it is equal to its biconjugate. Applying this property, we have

$$\begin{aligned} & \sup_{\boldsymbol{a}\in\mathbb{R}^d} \{g(b'\boldsymbol{a}^T\boldsymbol{w}) - \lambda\|\boldsymbol{a}-\boldsymbol{a}'\|\} \\ = & \sup_{\substack{\boldsymbol{a}\in\mathbb{R}^d, \\ \theta\in\mathrm{dom}(g^*)}} \{\theta b'\boldsymbol{a}^T\boldsymbol{w} - g^*(\theta) - \lambda\|\boldsymbol{a}-\boldsymbol{a}'\|\} \\ = & \sup_{\substack{\boldsymbol{a}\in\mathbb{R}^d, \\ \theta\in\mathrm{dom}(g^*)}} \{\theta b'(\boldsymbol{a}-\boldsymbol{a}')^T\boldsymbol{w} + \theta b'(\boldsymbol{a}')^T\boldsymbol{w} - g^*(\theta) - \lambda\|\boldsymbol{a}-\boldsymbol{a}'\|.\} \end{aligned}$$

Applying the change of variable $\boldsymbol{v} := \boldsymbol{a} - \boldsymbol{a}'$, we further have

$$\begin{aligned} & \sup_{\boldsymbol{a}\in\mathbb{R}^d} \{g(b'\boldsymbol{a}^T\boldsymbol{w}) - \lambda\|\boldsymbol{a}-\boldsymbol{a}'\|\} \\ = & \sup_{\substack{\boldsymbol{v}\in\mathbb{R}^d, \\ \theta\in\mathrm{dom}(g^*)}} \{\theta b'\boldsymbol{v}^T\boldsymbol{w} + \theta b'(\boldsymbol{a}')^T\boldsymbol{w} - g^*(\theta) - \lambda\|\boldsymbol{v}\|\} \\ = & \sup_{\theta\in\mathrm{dom}(g^*)} \{\mathbb{1}_{\{\|\theta b'\boldsymbol{w}\|_*\leq\lambda\}} + \theta b'(\boldsymbol{a}')^T\boldsymbol{w} - g^*(\theta)\} \\ = & \begin{cases} g(b'(\boldsymbol{a}')^T\boldsymbol{w}), & \text{if } \sup_{\theta\in\mathrm{dom}(g^*)}\|\theta b'\boldsymbol{w}\|_* \leq \lambda \\ +\infty, & \text{otherwise} \end{cases}, \end{aligned}$$

where the second equality is by the convex conjugate of a norm $\|\cdot\|$ being equal to the convex indicator of the unit ball w.r.t. the dual norm $\|\cdot\|_*$. Finally, it remains to use that, by Assumption 13, $\sup_{\theta\in\mathrm{dom}(g^*)}|\theta| = M$. □

Then following [SAMEK15, Theorem 1], we provide reformulations of the problem from Eq. (4.73) that can be addressed by computationally efficient solvers.

Theorem 11. *The optimization problem from Eq.* (4.73) *is equivalent to:*

$$\begin{aligned}
\min_{\boldsymbol{w},\lambda,\boldsymbol{u},\boldsymbol{v},\boldsymbol{s},\boldsymbol{t}} \quad & \rho\lambda + \frac{1}{n}\sum_{i=1}^{n} s_i \\
\text{s.t.} \quad & u_i = b_i \boldsymbol{a}_i^T \boldsymbol{w}, && i \in [n], \\
& v_i = -u_i, && i \in [n], \\
& t_i = 2\kappa\lambda + s_i, && i \in [n], \\
& g(u_i) \leq s_i, && i \in [n], \\
& g(v_i) \leq t_i, && i \in [n], \\
& \|\boldsymbol{w}\|_* \leq \lambda/M.
\end{aligned} \tag{4.77}$$

Proof. Let $\boldsymbol{z} = (\boldsymbol{a}, b) \in \Theta := \mathbb{R}^d \times \{1, -1\}$ and let $h_{\boldsymbol{w}}(\boldsymbol{z}) := g(b\boldsymbol{a}^T\boldsymbol{w})$. Then by the definition of the Wasserstein metric,

$$\sup_{\mathbb{P}\in\mathcal{P}_{\rho,\kappa}} \mathbb{E}^{\mathbb{P}}[g(b\boldsymbol{a}^T\boldsymbol{w})] = \begin{cases} \sup_{\pi\in\Pi(\mathbb{P},\hat{\mathbb{P}}_n)} & \int_\Theta h_{\boldsymbol{w}}(\boldsymbol{z})\pi(\mathrm{d}\boldsymbol{z},\Theta) \\ \text{s.t.} & \int_{\Theta\times\Theta} \zeta(\boldsymbol{z},\boldsymbol{z}')\pi(\mathrm{d}\boldsymbol{z},\mathrm{d}\boldsymbol{z}') \leq \rho. \end{cases} \tag{4.78}$$

Assume that the conditional distribution of $\boldsymbol{z}$ given $\boldsymbol{z}' = (\boldsymbol{a}_i, b_i)$ is $\mathbb{P}^i$, for all $i \in [n]$. Then, based on the definition of $\mathbb{P}_n = \frac{1}{n}\sum_{i=1}^n \delta_{(\boldsymbol{a}_i,b_i)}$, we have

$$\pi(\mathrm{d}\boldsymbol{z},\mathrm{d}\boldsymbol{z}') = \frac{1}{n}\sum_{i=1}^{n} \delta_{(\boldsymbol{a}_i,b_i)}\mathbb{P}^i(\mathrm{d}\boldsymbol{z}). \tag{4.79}$$

As a result, the problem from Eq. (4.78) is equivalent to

$$\sup_{\mathbb{P}\in\mathcal{P}_{\rho,\kappa}} \mathbb{E}^{\mathbb{P}}[g(b\boldsymbol{a}^T\boldsymbol{w})] = \begin{cases} \sup_{\mathbb{P}^i} & \frac{1}{n}\sum_{i=1}^n \int_\Theta h_{\boldsymbol{w}}(\boldsymbol{z})\mathbb{P}^i(\mathrm{d}\boldsymbol{z}) \\ \text{s.t.} & \frac{1}{n}\sum_{i=1}^n \int_\Theta \zeta(\boldsymbol{z},\boldsymbol{z}')\mathbb{P}^i(\mathrm{d}\boldsymbol{z}) \leq \rho \\ & \int_\Theta \mathbb{P}^i(\mathrm{d}\boldsymbol{z}) = 1. \end{cases} \tag{4.80}$$

Then substituting in $\boldsymbol{z} = (\boldsymbol{a}, b)$, using that the domain of y is $\{1, -1\}$, and decomposing $\mathbb{P}^i$ into unnormalized measures $\mathbb{P}_{\pm 1}(\mathrm{d}\boldsymbol{a}) = \mathbb{P}^i(\mathrm{d}\boldsymbol{a}, \{b = \pm 1\})$ supported on $\mathbb{R}^d$, the RHS of Eq. (4.80) can be simplified to

$$\begin{aligned}
\sup_{\mathbb{P}^i_\pm} \quad & \frac{1}{n}\sum_{i=1}^{n} \int_{\mathbb{R}^d} (h_{\boldsymbol{w}}(\boldsymbol{a}, 1)\mathbb{P}^i_1(\mathrm{d}\boldsymbol{a}) + h_{\boldsymbol{w}}(\boldsymbol{a}, -1)\mathbb{P}^i_{-1}(\mathrm{d}\boldsymbol{a})) \\
\text{s.t.} \quad & \frac{1}{n}\sum_{i=1}^{n} \int_{\mathbb{R}^d} (\zeta((\boldsymbol{a}, 1), (\boldsymbol{a}_i, \boldsymbol{b}_i))\mathbb{P}^i_1(\mathrm{d}\boldsymbol{a}) + \zeta((\boldsymbol{a}, -1), (\boldsymbol{a}_i, \boldsymbol{b}_i))\mathbb{P}^i_{-1}(\mathrm{d}\boldsymbol{a})) \leq \rho \\
& \int_{\mathbb{R}^d} (\mathbb{P}^i_1(\mathrm{d}\boldsymbol{a}) + \mathbb{P}^i_{-1}(\mathrm{d}\boldsymbol{a})) = 1.
\end{aligned}$$

With the definition of the cost function $\zeta((\boldsymbol{a},b),(\boldsymbol{a}',b'))=\|\boldsymbol{a}-\boldsymbol{a}'\|+\kappa|b-b'|$, it follows that

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\int_{\mathbb{R}^d}\zeta((\boldsymbol{a},1),(\boldsymbol{a}_i,\boldsymbol{b}_i))\mathbb{P}_1^i(\mathrm{d}\boldsymbol{a})+\zeta((\boldsymbol{a},-1),(\boldsymbol{a}_i,\boldsymbol{b}_i))\mathbb{P}_{-1}^i(\mathrm{d}\boldsymbol{a})\\
=&\frac{1}{n}\int_{\mathbb{R}^d}\sum_{b_i=1}[\|\boldsymbol{a}-\boldsymbol{a}_i\|\mathbb{P}_1^i(\mathrm{d}\boldsymbol{a})+\|\boldsymbol{a}-\boldsymbol{a}_i\|\mathbb{P}_{-1}^i(\mathrm{d}\boldsymbol{a})+2\kappa\mathbb{P}_{-1}^i(\mathrm{d}\boldsymbol{a})]\\
&+\frac{1}{n}\int_{\mathbb{R}^d}\sum_{b_i=-1}[\|\boldsymbol{a}-\boldsymbol{a}_i\|\mathbb{P}_{-1}^i(\mathrm{d}\boldsymbol{a})+\|\boldsymbol{a}-\boldsymbol{a}_i\|\mathbb{P}_1^i(\mathrm{d}\boldsymbol{a})+2\kappa\mathbb{P}_1^i(\mathrm{d}\boldsymbol{a})]\\
=&\frac{2\kappa}{n}\int_{\mathbb{R}^d}\Big(\sum_{b_i=1}\mathbb{P}_{-1}^i(\mathrm{d}\boldsymbol{a})+\sum_{b_i=-1}\mathbb{P}_1^i(\mathrm{d}\boldsymbol{a})\Big)+\frac{1}{n}\int_{\mathbb{R}^d}\sum_{i=1}^{n}\|\boldsymbol{a}-\boldsymbol{a}_i\|(\mathbb{P}_{-1}^i(\mathrm{d}\boldsymbol{a})+\mathbb{P}_1^i(\mathrm{d}\boldsymbol{a})).
\end{aligned}\tag{4.81}$$

Thus, we have

$$\sup_{\mathbb{P}\in\mathcal{P}_{\rho,\kappa}}\mathbb{E}^{\mathbb{P}}[g(b\boldsymbol{a}^T\boldsymbol{w})]=\begin{cases}
\sup\limits_{\mathbb{P}_{\pm1}^i}\dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\int_{\mathbb{R}^d}(h_{\boldsymbol{w}}(\boldsymbol{a},1)\mathbb{P}_1^i(\mathrm{d}\boldsymbol{a})+h_{\boldsymbol{w}}(\boldsymbol{a},-1)\mathbb{P}_{-1}^i(\mathrm{d}\boldsymbol{a}))\\
\text{s.t.}\quad \dfrac{2\kappa}{n}\displaystyle\int_{\mathbb{R}^d}\left(\sum_{b_i=1}\mathbb{P}_{-1}^i(\mathrm{d}\boldsymbol{a})+\sum_{b_i=-1}\mathbb{P}_1^i(\mathrm{d}\boldsymbol{a})\right)\\
\qquad\qquad+\dfrac{1}{n}\displaystyle\int_{\mathbb{R}^d}\sum_{i=1}^{n}\|\boldsymbol{a}-\boldsymbol{a}_i\|(\mathbb{P}_{-1}^i(\mathrm{d}\boldsymbol{a})+\mathbb{P}_1^i(\mathrm{d}\boldsymbol{a}))\le\rho\\
\qquad\displaystyle\int_{\mathbb{R}^d}(\mathbb{P}_1^i(\mathrm{d}\boldsymbol{a})+\mathbb{P}_{-1}^i(\mathrm{d}\boldsymbol{a}))=1.
\end{cases}$$

The above problem w.r.t. $\mathbb{P}_{\pm1}^i$ is an infinite-dimensional linear program with a finite number of constraints. By [Sha01, Proposition 3.4], we get the following equivalent dual formulation:

$$\sup_{\mathbb{P}\in\mathcal{P}_{\rho,\kappa}}\mathbb{E}^{\mathbb{P}}[g(b\boldsymbol{a}^T\boldsymbol{w})]=\begin{cases}
\min\limits_{\lambda,s_i}\rho\lambda+\dfrac{1}{n}\displaystyle\sum_{i=1}^{n}s_i\\
\text{s.t.}\ \sup\limits_{\boldsymbol{a}\in\mathbb{R}^d}h_{\boldsymbol{w}}(\boldsymbol{a},1)-\lambda\|\boldsymbol{a}-\boldsymbol{a}_i\|-\lambda\kappa(1-b_i)\le s_i,\quad i\in[n]\\
\qquad\sup\limits_{\boldsymbol{a}\in\mathbb{R}^d}h_{\boldsymbol{w}}(\boldsymbol{a},-1)-\lambda\|\boldsymbol{a}-\boldsymbol{a}_i\|-\lambda\kappa(1+b_i)\le s_i,\ i\in[n]\\
\qquad\lambda\ge0.
\end{cases}$$

Then, recalling that $h_{\boldsymbol{w}}(\boldsymbol{a},\pm1)$ is short for $g(\pm\boldsymbol{a}^T\boldsymbol{w})$, by Lemma 35, we have

$$\sup_{\boldsymbol{a}\in\mathbb{R}^d}h_{\boldsymbol{w}}(\boldsymbol{a},\pm1)-\lambda\|\boldsymbol{a}-\boldsymbol{a}_i\|=\begin{cases}
g(\pm\boldsymbol{a}_i^T\boldsymbol{w}), & \text{if } \sup_{\theta\in\mathrm{dom}(g^*)}\|\theta\boldsymbol{w}\|_*\le\lambda\\
+\infty, & \text{otherwise.}
\end{cases}$$

Finally, the resulting reformulation is

$$\begin{aligned} \min_{\boldsymbol{w},\lambda,\boldsymbol{s}} \ & \rho\lambda + \frac{1}{n}\sum_{i=1}^{n} s_i \\ \text{s. t.} \quad & g(b_i \boldsymbol{a}_i^T \boldsymbol{w}) \le s_i, \quad i \in [n], \\ & g(-b_i \boldsymbol{a}_i^T \boldsymbol{w}) - 2\lambda\kappa \le s_i, \quad i \in [n], \\ & \sup_{\theta \in \mathrm{dom}(g^*)} \|\theta \boldsymbol{w}\|_* \le \lambda. \end{aligned}$$

Finally, recalling that, by assumption, $\sup_{\theta\in\mathrm{dom}(g^*)}|\theta| = M$, it follows that the constraint $\sup_{\theta\in\mathrm{dom}(g^*)}\|\theta\boldsymbol{w}\|_* \le \lambda$ is equivalent to $\|\boldsymbol{w}\|_* \le \lambda/M$. Meanwhile, by introducing $u_i = b_i \boldsymbol{a}_i^T \boldsymbol{w}, v_i = -u_i (i \in [n])$ and $s_i, t_i (i \in [n]$, we obtain Theorem 11. □

In Theorem 11, when we assume that the conic constraints $g(u) \le s$ and $\|\boldsymbol{w}\|_* \le \lambda/M$ in Eq. (4.77) admit efficient proximal operators, we can formulate this problem as (PD-GLP) and apply CLVR. The resulting complexity bounds are similar to those discussed above for the f-divergence formulation.

4.6 Numerical experiments

We provide experimental evaluations of our algorithm for the reformulation of the DRO with Wasserstein metric based on the ℓ_1-norm (with $\kappa = 0.1$ and $\rho = 10$) and hinge loss. For its LP formulation (see Theorem 11), we compare our CLVR method with three representative methods: PDHG [CP11], SPDHG [CERS18] and PURE-CD [AFC20]. For all algorithms we use LPMetric (4.45) as the performance measure and use a restart strategy based on successive halving of LPMetric (Section 4.4.3) to obtain linear convergence. We implemented CLVR and other algorithms in Julia, optimizing all implementations to the extent possible. Our code is available at `https://github.com/ericlincc/Efficient-GLP`.

Comparison between values of L and R. As described in Section 4.1, a major advantage of CLVR is that the complexity of CLVR depends on the max row norm R instead of the spectral norm L, which in the worst case for ill-conditioned problems can lead to a factor of $\sqrt{n}$ improvement. In practical problems where the problem instances are highly structured (e.g., reformulated DRO problems), R can be much smaller than L. Table 4.2 provides empirical evidence for this claim. In all our experiments, we normalize each rows of $\boldsymbol{A}$ to $R = 1$ as stated in Assumption 9, so the values of L demonstrate the theoretical improvements for the experiments described in Section 4.6.

Comparison with primal-dual algorithms. Figure 4.1 provides a comparison between algorithms in terms of the number of data passes and wall-clock time. The spikes in all the plots are due

Table 4.2: Values of the spectral norm L in the reformulated DRO problems with Wasserstein metric after each row is normalized to $R = 1$.

Reformulated a9a	Reformulated gisette	Reformulated rcv1	Reformulated news20
$d = 130738, n = 97929$	$d = 44002, n = 28000$	$d = 269914, n = 155198$	$d = 5500750, n = 2770370$
117.3	65.9	196.4	1041.6

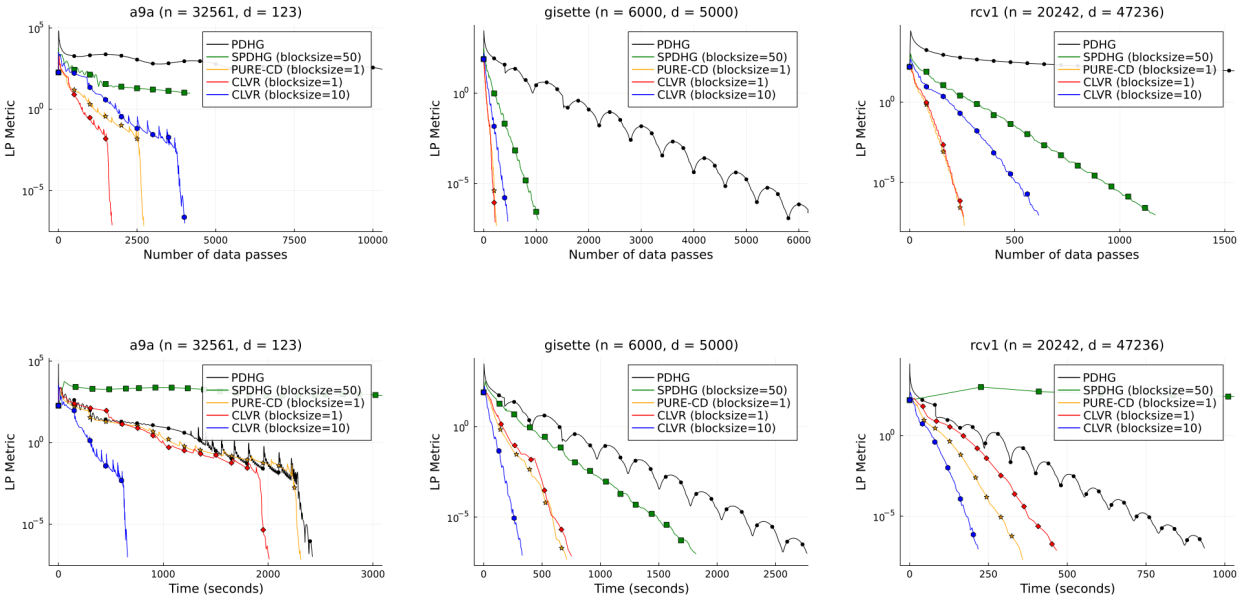


Figure 4.1: Comparison of numerical results in terms of number of data passes and wall-clock time.

to restarts: At the beginning of each restart cycle, the value of LPMetric increases significantly, then decreases rapidly. For the number of data passes (top row), CLVR with block size 1 and PURE-CD perform best on all three datasets, CLVR with block size 10 and SPDHG with block size 50 have second-tier performance, and PDHG is worst. For the CLVR algorithm, smaller block size corresponds to smaller $\hat{L}$ in Assumption 11, which corresponds to better complexity in terms of data passes by Theorem 8. Nevertheless, the gap between empirical performance and theoretical guarantee for SPDHG and PURE-CD deserves further research because, to date, they have only been shown to have the same iteration complexity as PDHG. [5] Empirically, on `a9a`, CLVR with block size 1 performs better than PURE-CD in terms of data passes.

In terms of wall-clock time (bottom row of Figure 4.1), because of different per-iteration costs of each algorithm and instruction-level parallelism in modern processors [HP11], the plots differ

[5] The later paper [ACW22] describes complexity results for a newly developed version of PURE-CD that exploits sparsity in $\boldsymbol{A}$.

significantly from the plots for number of data passes. Even with block size 50, SPDHG spends the most wall-clock time for one data pass and is the slowest on sparse datasets `a9a` and `rcv1`, but is faster than PDHG on the dense dataset `gisette`. Meanwhile, while CLVR with block size 10 is not best in terms of data passes, it remains fastest in terms of wall-clock time on all datasets due to cheaper per-iteration cost and instruction-level parallelism. On `rcv1`, the per-iteration cost of PURE-CD is about 60% of that of CLVR with block size 1. Hence, despite having similar performance in terms of data passes, PURE-CD is faster than CLVR with block size 1, but is still slower than CLVR with block size 10.

Comparison with production linear programming solvers. Table 4.3 shows that CLVR is competitive against production-quality linear programming solvers such as GLPK [glp22] and Gurobi [Gur22]. We observe that CLVR reached accurate solutions significantly faster than GLPK and Gurobi in the reformulated problems with `gisette` and `rcv1` datasets. Although CLVR is much slower than Gurobi(barrier) on `a9a` dataset, we believe that much of the performance gap in this case is due to the redundancy in the problem formulation with the `a9a` dataset, much of which is removed by Gurobi presolver[6]. We leave presolving and other heuristic speedups of CLVR for future work.

Table 4.3: Comparison of numerical results between CLVR and three production solvers for linear programming, showing time required (in seconds) for each solver to reach accuracy 10^{-8}.

Time (seconds)	Reformulated a9a $d = 130738, n = 97929$	Reformulated gisette $d = 44002, n = 28000$	Reformulated rcv1 $d = 269914, n = 155198$
JuMP+GLPK	899	$> 4 \times 10^4$	$> 4 \times 10^4$
JuMP+Gurobi(simplex)	893	2482	7008
JuMP+Gurobi(barrier)	**26**	1039.7	1039.5
CLVR	962	**697**	**582**

4.6.1 Comparison of adaptive restart schemes

We provide a brief empirical comparison between our adaptive restart scheme that uses LPMetric and the adaptive restart scheme using the normalized duality gap proposed in [AHLL21]. We compared the performance of PDHG on benchmark problem sets `qap10`, `qap15`, `nug08`, and `nug20` used in [AHLL21], using the two adaptive restart criteria. We ran PDHG until reaching accuracy as

[6]In our DRO instance with `a9a` dataset, Gurobi presolver removed 25% of the columns and 58% of the nonzeros.

described in [AHLL21] (that is, until normalized duality gap is at most 10^{-6} and primal and dual infeasibility is at most 10^{-8}).

Table 4.4: Number of iterations required for the normalized duality gap and primal and dual infeasibility to fall below 10^{-6} and 10^{-8}, respectively.

Problem Name	Adaptive Normalized Duality Gap	Adaptive LPMetric
`qap10`	**13041**	14521
`qap15`	12561	**961**
`nug08`	**841**	1481
`nug20`	22001	**16281**

Table 4.4 shows that the two restart criteria give similar performance in terms of iteration complexity. Normalized gap is better on `qap10` and `nug08`, while LPMetric is better on `qap15` and `nug20`. For further details, Figure 4.2 plots the normalized duality gap vs iteration count. The two adaptive restart schemes lead to similar performance of PDHG over iterations. Comparisons based on wall-clock time are shown in Figure 4.3; the behavior is similar. We conclude that our restart criterion based on LPMetric seems comparable with normalized duality gap, in terms of iteration complexity.

4.6.2 Batch optimizations for practical computations

When we consider the DRO problem with Wasserstein metric of ℓ_1-norm and with hinge loss, we observe that the reformulation described in Theorem 11 can be further reformulated into an ordinary LP, as the dual norm of ℓ_1 norm is ℓ_∞ norm and hinge loss can be decomposed linearly with additional auxiliary variables. Thus in the following instances we consider, we apply our adaptive restart scheme with respect to LPMetric as illustrated in Section 4.4.3 to achieve heuristic linear convergence rate in terms of the number of data passes. We compare our CLVR method with three representative methods: PDHG [CP11], SPDHG [CERS18] and PURE-CD [AFC20]. We implemented CLVR and other algorithms in Julia, optimizing all implementations to the best of our ability.[7] For SPDHG, whose per-iteration cost is at least $O(d)$, we consider a large batch size of 50 to balance the effect of the $O(d)$ cost and improve the overall efficiency. Meanwhile, PURE-CD with block size 1 is already well suited to sparsity. For CLVR, we experiment with block sizes 1 and 10.

We conducted our experiments on LibSVM [CL11] datasets `a9a`, `gisette`, and `rcv1.binary`, each with different sparsity levels. We run each algorithm using one CPU core, on a Linux machine with a second generation Intel Xeon Scalable Processor (Cascade Lake-SP) with 128 GB of RAM. Because the weight parameter γ between primal and dual variables (see Theorem 8) strongly influences

[7] Julia is particularly designed for high performance numerical computation.

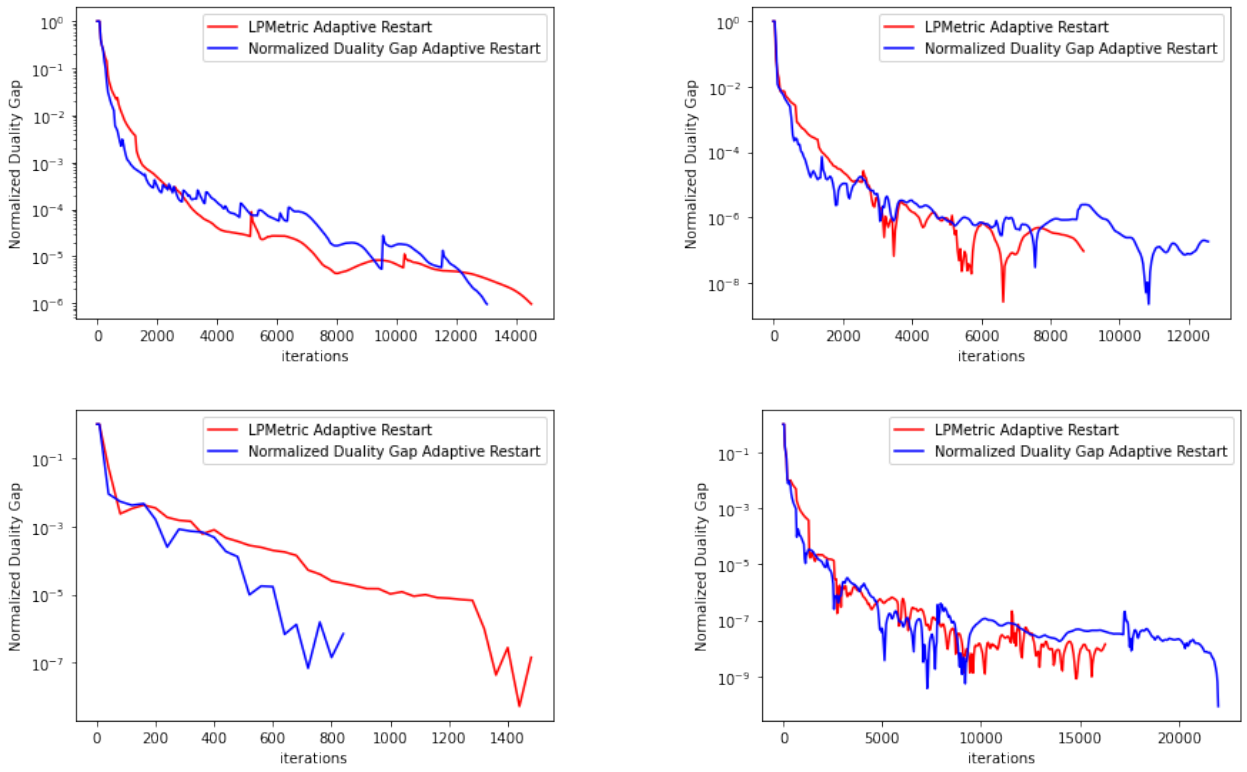


Figure 4.2: Comparisons of restart schemes that use LPMetric and that use the normalized duality gap against number of iterations. The plots from left to right and then top to bottom are for `qap10`, `qap15`, `nug08`, and `nug20`.

empirical performance, we tune it for all datasets by trying the values $\{10^{-i}\}$ for $i \in \mathbb{Z}$, for each of the methods. We set the Lipschitz constant of PDHG to be the largest singular value of the constraint matrix in the LP formulation. For PURE-CD and CLVR with block size 1, because the rows of the matrix are normalized, we set the Lipschitz constant to 1. For CLVR with block size 10 and SPDHG with block size 50, the Lipschitz constants are tuned to 3 and 9, respectively.

Remark 4 (Comparisons of using different block sizes)**.** *We conducted experiments to compare the practical performance of* CLVR *against different choices of block sizes, with results shown in Figure 4.4. We ran the DRO with Wasserstein metric using the same setup as described in Section 4.6, on the* `rcv1` *dataset and using an early stopping criterion of* `LPMetric` *at* 10^{-1}*. In the plot, we can see that* CLVR *converges to an approximate solution fastest when the block size is set to* 10*, providing support for our choice of* 10 *in Section 4.6. As illustrated in Figure 4.1,* CLVR *is most efficient in terms of the number of data passes when the block size is 1, but in terms of the execution time, running* CLVR *with larger block size yields better performance. We attribute this phenomenon to the instruction-level*

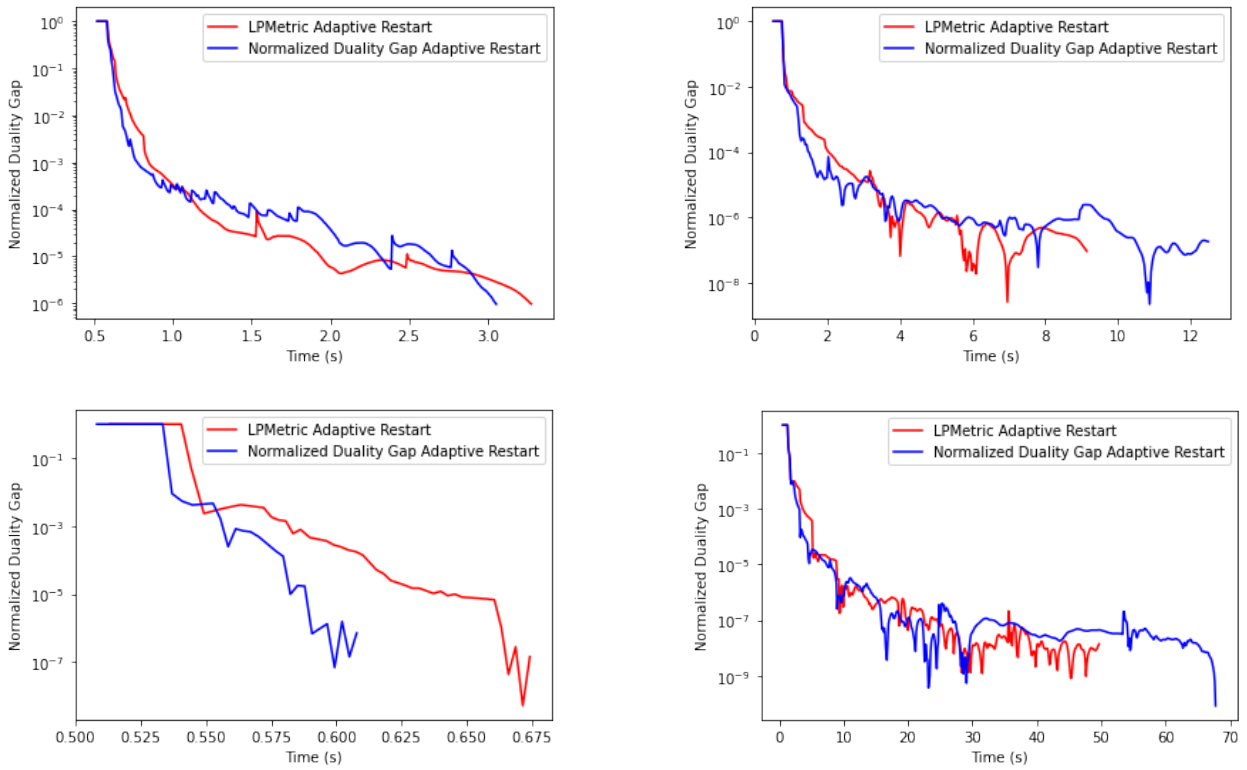


Figure 4.3: Comparisons of restart schemes that use LPMetric and that use the normalized duality gap against wall clock time in terms of seconds. The plots from left to right and then top to bottom are for `qap10`, `qap15`, `nug08`, and `nug20`.

parallelism [HP11] in modern processors, allowing more computations to be completed in the same number of clock cycles.

In Table 4.5, we list information about the three datasets and the corresponding matrices in the reformulations. As we see, due to the sparse connectivity of auxiliary variables, all the matrices in reformulations are quite sparse. As a commonly adopted preprocessing step for LP, we normalize the matrix in the standard-form LP so that each row has Euclidean norm 1.

Remark 5 (Performance comparison using multiple cores)**.** *We conducted further experiments to examine the effects of allowing the algorithms to run on more computing cores. However, we did not observe any meaningful difference in performances in terms of wall-clock time when we repeated the experiments described above and in Section 4.6 using 2 CPU cores per algorithm. Our interpretation of this observation is that because most of the steps within* CLVR *and other algorithms we are comparing against are simple and cheap, involving very few large matrix-vector multiplications*

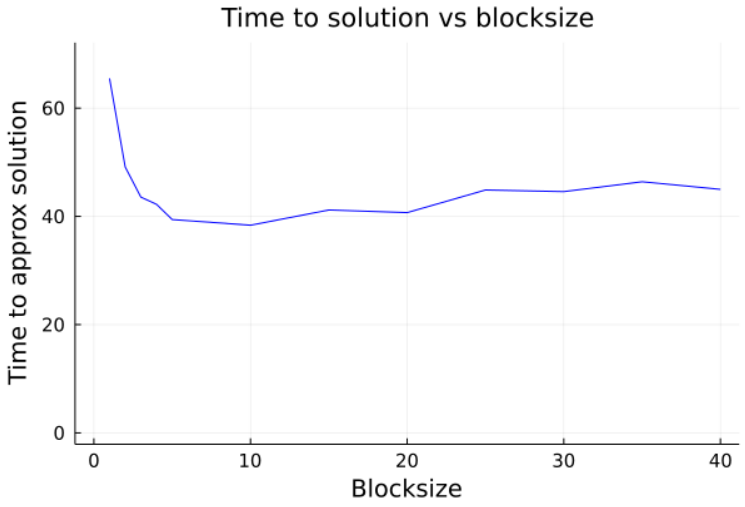


Figure 4.4: Comparison of the performance of CLVR with various choices of blocksizes.

and no matrix factorization, the practical performance of algorithms becomes memory-bound, hence additional cores do not make much difference.

Conclusion. Our numerical experiments show that CLVR is fastest in both the number of data passes and wall-clock time on considered datasets, among all primal-dual algorithms that we implemented. It is also competitive with production-quality linear programming solvers. Since it has a theoretical guarantee that matches or improves the state of the art among primal-dual methods, we believe that CLVR could be a method of choice.

Table 4.5: The dimension and sparsity of the original datasets and the corresponding matrices in reformulations.

Dataset	Original (d, n)	#nonzeros / $(d \times n)$	Reformulated (d, n)	#nonzeros / $(d \times n)$
`a9a`	$(123, 32561)$	0.11	$(130738, 97929)$	9.6×10^{-5}
`gisette`	$(5000, 6000)$	0.99	$(44002, 28000)$	4.9×10^{-2}
`rcv1`	$(47236, 20242)$	1.5×10^{-3}	$(269914, 155198)$	8.8×10^{-5}

Chapter 5

Conclusions and Future Directions

In this dissertation, we designed and analyzed coordinate algorithms for solving modern structured machine learning problems under some favorable structures that arises in large-scale settings, given in Eq. (1.1). Our results contribute to the algorithmic understandings of continuous first-order optimization via efficient and scalable coordinate algorithms. Below we highlight general conclusions and possible future directions.

Chapter 2: Tighter Convergence Bounds for Shuffled SGD via Primal-Dual Perspective. Through the lens of primal-dual perspective, we provided the first fine-grained, data-dependent convergence bounds for shuffled SGD, a class of incremental method with random permutations, in general convex finite-sum and in generalized linear model settings. There remains further explorations to be done to extend our results to less favorable settings such as loss functions with non-convexity with mild assumptions, which can generally appear in modern deep learning settings.

Chapter 3: Accelerating Cyclic Coordinate Algorithms via Dual Averaging with Extrapolation. We further extended the cyclic smoothness idea from [SD21a] to propose the A-CODER algorithm 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 showed that A-CODER attains the optimal convergence rate with improved dependence on the number of blocks compared to prior work. We demonstrated that the new dependence on the number of blocks may be of constant factors for many machine learning datasets through numerical experiments, despite that dependency being the order of the square root of the number of blocks in the worst case. As a note for future direction, there still remains a gap in our theoretical understanding about what structures should present in the objective functions for that extra dependency on the number of blocks to diminish fully.

Chapter 4: Faster Algorithms for Solving Generalized Linear Programming and the Connection to Distributionally Robust Optimization. We studied a class of generalized linear programs (GLP) in a large-scale setting, which includes a simple, possibly non-smooth convex regularizer and simple convex set constraints. By reformulating (GLP) as an equivalent convex-concave min-max problem, we designed an efficient, scalable first-order algorithm named Coordinate Linear Variance Reduction (CLVR). CLVR yielded improved complexity results for (GLP) that depend on the max row norm of the linear constraint matrix $\boldsymbol{A}$ rather than the spectral norm. We further introduced two strategies to improve the convergence rates: 1) lazy updates when the regularization term and constraints are coordinate-separable, and 2) an adaptive restart scheme when the regularization term is zero. Furthermore, by introducing sparsely connected auxiliary variables, we showed that Distributionally Robust Optimization (DRO) problems with ambiguity sets based on both f-divergence and Wasserstein metrics can be reformulated as (GLPs).

LIST OF REFERENCES

[A+17] M.G. Aartsen et al. The IceCube neutrino observatory: instrumentation and online systems. *Journal of Instrumentation*, 12(03):P03012–P03012, mar 2017.

[AA00] Erling D Andersen and Knud D Andersen. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High Performance Optimization*, pages 197–232. Springer, 2000.

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[ACW22] Ahmet Alacaoglu, Volkan Cevher, and Stephen J. Wright. On the complexity of a practical primal-dual coordinate method, 2022.

[ADFC17] Ahmet Alacaoglu, Quoc Tran Dinh, Olivier Fercoq, and Volkan Cevher. Smooth primal-dual coordinate descent algorithms for nonsmooth convex optimization. In *Proc. NIPS'17*, 2017.

[ADH+21] David Applegate, Mateo Díaz, Oliver Hinder, Haihao Lu, Miles Lubin, Brendan O'Donoghue, and Warren Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. *Proc. NeurIPS'21*, 2021.

[AFC19] Ahmet Alacaoglu, Olivier Fercoq, and Volkan Cevher. On the convergence of stochastic primal-dual hybrid gradient. *arXiv preprint arXiv:1911.00799*, 2019.

[AFC20] Ahmet Alacaoglu, Olivier Fercoq, and Volkan Cevher. Random extrapolation for primal-dual coordinate descent. In *Proc. ICML'20*, 2020.

[AHLL21] David Applegate, Oliver Hinder, Haihao Lu, and Miles Lubin. Faster first-order primal-dual methods for linear programming using restarts and sharpness. *arXiv preprint arXiv:2105.12715*, 2021.

[AM22] Ahmet Alacaoglu and Yura Malitsky. Stochastic variance reduction for variational inequality methods. In *Conference on Learning Theory*, pages 778–816, 2022.

[AWBR09] Alekh Agarwal, Martin J Wainwright, Peter Bartlett, and Pradeep Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Proc. NeurIPS'09*, 2009.

[AYS20] Kwangjun Ahn, Chulhee Yun, and Suvrit Sra. SGD with shuffling: optimal rates without component convexity and large epoch requirements. In *Proc. NeurIPS'20*, 2020.

[AZ17] Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proc. ACM STOC'17*, 2017.

[AZQRY16] Zeyuan Allen-Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *Proc. ICML'16*, 2016.

[B+15] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 2015.

[BCN18] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

[BDBC+10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[Bec17] Amir Beck. *First-order methods in optimization*. SIAM, 2017.

[Ben12] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade: Second Edition*, 2012.

[BGK18] Dimitris Bertsimas, Vishal Gupta, and Nathan Kallus. Robust sample average approximation. *Mathematical Programming*, 171(1):217–282, 2018.

[BHK20] Avrim Blum, John Hopcroft, and Ravi Kannan. *Foundations of Data Science*. Cambridge University Press, Cambridge, 2020.

[Bia16] Pascal Bianchi. Ergodic convergence of a stochastic proximal point algorithm. *SIAM Journal on Optimization*, 26(4):2235–2260, 2016.

[Bot09] Léon Bottou. Curiously fast convergence of some stochastic gradient descent algorithms. Unpublished open problem offered to the attendance of the SLDS 2009 conference, 2009.

[BT09] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[BT13] Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[CDLP21] Alejandro Carderera, Jelena Diakonikolas, Cheuk Yin Lin, and Sebastian Pokutta. Parameter-free locally accelerated conditional gradients. In *International Conference on Machine Learning*, 2021.

[CDO18] Michael Cohen, Jelena Diakonikolas, and Lorenzo Orecchia. On acceleration with noise-corrupted gradients. In *Proc. ICML'18*, 2018.

[CERS18] Antonin Chambolle, Matthias J Ehrhardt, Peter Richtárik, and Carola-Bibiane Schonlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM Journal on Optimization*, 28(4):2783–2808, 2018.

[CG16] Jinghui Chen and Quanquan Gu. Accelerated stochastic block coordinate gradient descent for sparsity constrained nonconvex optimization. In *Conference on Uncertainty in Artificial Intelligence*, 2016.

[CJST19] Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Variance reduction for matrix games. In *Proc. NeurIPS'19*, 2019.

[CJST20] Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Coordinate methods for matrix games. *arXiv preprint arXiv:2009.08447*, 2020.

[CL11] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[CLD24] Xufeng Cai, Cheuk Yin Lin, and Jelena Diakonikolas. Tighter convergence bounds for shuffled sgd via primal-dual perspective. In *Advances in Neural Information Processing Systems*, volume 37, pages 72475–72524, 2024.

[CLY23] Jaeyoung Cha, Jaewook Lee, and Chulhee Yun. Tighter lower bounds for shuffling SGD: Random permutations and beyond. *arXiv preprint arXiv:2303.07160*, 2023.

[CP11] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[CSWD22a] Xu Cai, Chaobing Song, Stephen J. Wright, and Jelena Diakonikolas. Cyclic block coordinate descent with variance reduction for composite nonconvex optimization. *ArXiv*, abs/2212.05088, 2022.

[CSWD22b] Xufeng Cai, Chaobing Song, Stephen J Wright, and Jelena Diakonikolas. Cyclic block coordinate descent with variance reduction for composite nonconvex optimization. *arXiv preprint arXiv:2212.05088*, 2022.

[CW17] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[CWY17] Yat Tin Chow, Tianyu Wu, and Wotao Yin. Cyclic coordinate-update algorithms for fixed-point problems: Analysis and applications. *SIAM Journal on Scientific Computing*, 39(4):A1280–A1300, 2017.

[DBLJ14] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

[Den12] Li Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[DFVR03] Daniela Pucci De Farias and Benjamin Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

[DG21] Jelena Diakonikolas and Cristóbal Guzmán. Complementary composite minimization, small gradients in general norms, and applications to regression problems. *arXiv preprint arXiv:2101.11041*, 2021.

[DGN21] John C Duchi, Peter W Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 2021.

[DL15] Cong D Dang and Guanghui Lan. Stochastic block mirror descent methods for nonsmooth and stochastic optimization. *SIAM Journal on Optimization*, 25(2):856–881, 2015.

[DN19] John Duchi and Hongseok Namkoong. Variance-based regularization with convex objectives. *The Journal of Machine Learning Research*, 20(1):2450–2504, 2019.

[DN21] John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *Annals of Statistics*, 49(3):1378–1406, 2021.

[DO18] Jelena Diakonikolas and Lorenzo Orecchia. Alternating randomized block coordinate descent. In *Proc. ICML'18*, 2018.

[DO19] Jelena Diakonikolas and Lorenzo Orecchia. The approximate duality gap technique: A unified theory of first-order methods. *SIAM Journal on Optimization*, 29(1):660–689, 2019.

[DR56] Jim Douglas and Henry H Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–439, 1956.

[DY10] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

[EK18] Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.

[FB19] Olivier Fercoq and Pascal Bianchi. A coordinate-descent primal-dual algorithm with large step size and possibly nonseparable functions. *SIAM Journal on Optimization*, 29(1):100–134, 2019.

[FHT10] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[FR15] Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

[glp22] GLPK (GNU linear programming kit). `https://www.gnu.org/software/glpk/`, 2022.

[GN18] Alexander Vladimirovich Gasnikov and Yu E Nesterov. Universal method for stochastic composite optimization problems. *Computational Mathematics and Mathematical Physics*, 58(1):48–64, 2018.

[GOP21] Mert Gürbüzbalaban, Asu Ozdaglar, and Pablo A Parrilo. Why random reshuffling beats stochastic gradient descent. *Mathematical Programming*, 186:49–84, 2021.

[GOPV17] Mert Gürbüzbalaban, Asuman Ozdaglar, Pablo A Parrilo, and N Denizcan Vanli. When cyclic coordinate descent outperforms randomized coordinate descent. In *Proc. NIPS'17*, 2017.

[GSS14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[Gur22] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022.

[GZ17] Xiang Gao and Shu-Zhong Zhang. First-order algorithms for convex optimization with nonseparable objective and coupled constraints. *Journal of the Operations Research Society of China*, 5(2):131–159, 2017.

[HJ20] Erfan Yazdandoost Hamedani and Afrooz Jalilzadeh. A stochastic variance-reduced accelerated primal-dual method for finite-sum saddle-point problems. *arXiv preprint arXiv:2012.13456*, 2020.

[HNSS18] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *Proc. ICML'18*, 2018.

[HNW22] N. Ho-Nguyen and S. J. Wright. Adversarial classification via distributional robustness with Wasserstein ambiguity. *Mathematical Programming, Series B*, 2022.

[Hof03] Alan J Hoffman. On approximate solutions of systems of linear inequalities. In *Selected Papers Of Alan J Hoffman: With Commentary*, pages 174–176. World Scientific, 2003.

[HP11] John L Hennessy and David A Patterson. *Computer architecture: a quantitative approach*. Elsevier, 2011.

[HS19] Jeff Haochen and Suvrit Sra. Random shuffling beats SGD after finite epochs. In *Proc. ICML'19*, 2019.

[JNT11] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[JZ13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Proc. NIPS'13*, 2013.

[KGB+16] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

[KGB17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017.

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Kor76] G. Korpelevich. The extragradient method for finding saddle points and other problems. *Ekonomika i Matematicheskie Metody*, 12(5):747–756 (in Russian; English translation in Matekon), 1976.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

[LAL+20] Changliu Liu, Tomer Arnon, Christopher Lazarus, Christopher Strong, Clark Barrett, Mykel J Kochenderfer, et al. Algorithms for verifying deep neural networks. *Foundations and Trends® in Optimization*, 4, 2020.

[LCDS20] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. In *Proc. NeurIPS'20*, 2020.

[LFP19] Puya Latafat, Nikolaos M Freris, and Panagiotis Patrinos. A new randomized block-coordinate primal-dual proximal algorithm for distributed optimization. *IEEE Transactions on Automatic Control*, 64(10):4050–4065, 2019.

[LHS19] Jiajin Li, Sen Huang, and Anthony Man-Cho So. A first-order algorithmic framework for Wasserstein distributionally robust logistic regression. *Proc. NeurIPS'19*, 2019.

[LJCJ17] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via SCSG methods. In *Proc. NIPS'17*, 2017.

[LLX15] Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization*, 25(4):2244–2273, 2015.

[LS18] Jinlong Lei and Uday V. Shanbhag. 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.

[LSC12] Vebjorn Ljosa, Katherine L. Sokolnicki, and Anne E Carpenter. Broad bioimage benchmark collection. `https://bbbc.broadinstitute.org/image_sets`, 2012. Accessed: 2023-05-16.

[LSD23] Cheuk Yin Lin, Chaobing Song, and Jelena Diakonikolas. Accelerated cyclic coordinate dual averaging with extrapolation for composite convex optimization. In *International Conference on Machine Learning*, 2023.

[LW12] Sangkyun Lee and Stephen J. Wright. Manifold identification in dual averaging for regularized stochastic online learning. *Journal of Machine Learning Research*, 13(6), 2012.

[LW19] Ching-Pei Lee and Stephen J Wright. Random permutations fix a worst case for cyclic coordinate descent. *IMA Journal of Numerical Analysis*, 39(3):1246–1275, 2019.

[LWR+14] Ji Liu, Steve Wright, Christopher Ré, Victor Bittorf, and Srikrishna Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. In *Proc. ICML'14*, 2014.

[LY21] Haihao Lu and Jinwen Yang. Nearly optimal linear convergence of stochastic primal-dual methods for linear programming. *arXiv preprint arXiv:2111.05530*, 2021.

[LZA+17] Xingguo Li, Tuo Zhao, Raman Arora, Han Liu, and Mingyi Hong. On faster convergence of cyclic block coordinate descent-type methods for strongly convex minimization. *The Journal of Machine Learning Research*, 18(1):6741–6764, 2017.

[Man84] Olvi L Mangasarian. Normal solutions of linear programs. In *Mathematical Programming at Oberwolfach II*, pages 206–216. Springer, 1984.

[Man04] OL Mangasarian. A Newton method for linear programming. *Journal of Optimization Theory and Applications*, 121(1):1–18, 2004.

[MFH11] Rahul Mazumder, Jerome H Friedman, and Trevor Hastie. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495):1125–1138, 2011.

[MKR20] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Random reshuffling: Simple analysis with vast improvements. In *Proc. NeurIPS'20*, 2020.

[MM79] Olvi L Mangasarian and RR Meyer. Nonlinear perturbation of linear programs. *SIAM Journal on Control and Optimization*, 17(6):745–752, 1979.

[MMS+18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[ND16] Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Proc. NIPS'16*, 2016.

[Nem04] Arkadi Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[Nes83] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2). In *Doklady AN USSR*, volume 269, pages 543–547, 1983.

[Nes07a] Yurii Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2-3):319–344, 2007.

[Nes07b] Yurii Nesterov. Gradient methods for minimizing composite objective function, 2007.

[Nes12] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[NJN19] Dheeraj Nagaraj, Prateek Jain, and Praneeth Netrapalli. SGD without replacement: Sharper rates for general smooth convex functions. In *Proc. ICML'19*, 2019.

[NTDP+21] Lam M Nguyen, Quoc Tran-Dinh, Dzung T Phan, Phuong Ha Nguyen, and Marten Van Dijk. A unified convergence analysis for shuffling-type gradient methods. *The Journal of Machine Learning Research*, 22(1):9397–9440, 2021.

[OHTG13] Hua Ouyang, Niao He, Long Tran, and Alexander Gray. Stochastic alternating direction method of multipliers. In *Proc. ICML'13*, 2013.

[OX19] Yuyuan Ouyang and Yangyang Xu. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *Mathematical Programming*, pages 1–35, 2019.

[PC+19] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[PN17] Andrei Patrascu and Ion Necoara. Nonasymptotic convergence of stochastic proximal point methods for constrained convex optimization. *Journal of Machine Learning Research*, 18(1):7204–7245, 2017.

[PWX+16] Zhimin Peng, Tianyu Wu, Yangyang Xu, Ming Yan, and Wotao Yin. Coordinate friendly structures, algorithms and applications. *ArXiv*, abs/1601.00863, 2016.

[RGP20] Shashank Rajput, Anant Gupta, and Dimitris Papailiopoulos. Closing the convergence gap of SGD without replacement. In *Proc. ICML'20*, 2020.

[RHS+16] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *Proc. ICML'16*, 2016.

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[RM19] Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A Review. *arXiv preprint arXiv:1908.05659*, 2019.

[Roc76] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

[RR13] Benjamin Recht and Christopher Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation*, 2013.

[RT16] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1-2):433–484, 2016.

[SAMEK15] Soroosh Shafieezadeh Abadeh, Peyman Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. In *Proc. NIPS'15*, 2015.

[SD21a] Chaobing Song and Jelena Diakonikolas. Cyclic coordinate dual averaging with extrapolation for generalized variational inequalities. *arXiv preprint arXiv:2102.13244*, 2021.

[SD21b] Chaobing Song and Jelena Diakonikolas. Fast cyclic coordinate dual averaging with extrapolation for generalized variational inequalities. *arXiv preprint arXiv:2102.13244*, 2021.

[Sha01] Alexander Shapiro. On duality theory of conic linear problems. In *Semi-Infinite Programming*, pages 135–165. Springer, 2001.

[Sha16] Ohad Shamir. Without-replacement sampling for stochastic gradient methods. In *Proc. NeurIPS'16*, 2016.

[SJ19] Matthew Staib and Stefanie Jegelka. Distributionally robust optimization and generalization in kernel methods. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[SJM20] Chaobing Song, Yong Jiang, and Yi Ma. Variance reduction via accelerated dual averaging for finite-sum optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

[SJM21] Chaobing Song, Yong Jiang, and Yi Ma. Unified acceleration of high-order algorithms under general hölder continuity. *SIAM Journal on Optimization*, 31(3):1797–1826, 2021.

[SLRB17] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, 2017.

[SLWD22] Chaobing Song, Cheuk Yin Lin, Stephen Wright, and Jelena Diakonikolas. Coordinate linear variance reduction for generalized linear programming. In *Advances in Neural Information Processing Systems*, 2022.

[SSZ13] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(1), 2013.

[ST13] Ankan Saha and Ambuj Tewari. On the nonasymptotic convergence of cyclic coordinate descent methods. *SIAM Journal on Optimization*, 23(1):576–601, 2013.

[Sto63] Josef Stoer. Duality in nonlinear programming and the minimax theorem. *Numerische Mathematik*, 5(1):371–379, 1963.

[SWD21] Chaobing Song, Stephen J Wright, and Jelena Diakonikolas. Variance reduction via primal-dual accelerated dual averaging for nonsmooth convex finite-sums. In *Proc. ICML'21*, 2021.

[SY19] Ruoyu Sun and Yinyu Ye. Worst-case complexity of cyclic coordinate descent: $O(n^2)$ gap with randomized version. *Mathematical Programming*, pages 1–34, 2019.

[SY21] Ruoyu Sun and Yinyu Ye. Worst-case complexity of cyclic coordinate descent: $O(n^2)$ gap with randomized version. *Mathematical Programming*, 185(1):487–520, 2021.

[TNTD21] Trang H Tran, Lam M Nguyen, and Quoc Tran-Dinh. SMG: A shuffling gradient-based method with momentum. In *Proc. ICML'21*, 2021.

[TSN22] Trang H Tran, Katya Scheinberg, and Lam M Nguyen. Nesterov accelerated shuffling gradient method for convex optimization. In *Proc. ICML'22*, 2022.

[Vap13] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[Vil09] Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2009.

[WKS14] Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

[WL+08] Tong Tong Wu, Kenneth Lange, et al. Coordinate descent algorithms for lasso penalized regression. *Annals of Applied Statistics*, 2(1):224–244, 2008.

[WL20] Stephen Wright and Ching-pei Lee. Analyzing random permutations for cyclic coordinate descent. *Mathematics of Computation*, 89(325):2217–2248, 2020.

[Wri15] Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[Xia10] Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

[Xu20] Yangyang Xu. Primal-dual stochastic gradient method for convex programs with many functional constraints. *SIAM Journal on Optimization*, 30(2):1664–1692, 2020.

[XY14] Yangyang Xu and Wotao Yin. Block stochastic gradient iteration for convex and nonconvex optimization. *ArXiv*, abs/1408.2597, 2014.

[XY15] Yangyang Xu and Wotao Yin. Block stochastic gradient iteration for convex and nonconvex optimization. *SIAM Journal on Optimization*, 25(3):1686–1716, 2015.

[XY17] Yangyang Xu and Wotao Yin. A globally convergent algorithm for nonconvex optimization based on block coordinate update. *Journal of Scientific Computing*, 72(2):700–734, 2017.

[YLMJ21] Yaodong Yu, Tianyi Lin, Eric Mazumdar, and Michael I Jordan. Fast distributionally robust learning with variance reduced min-max optimization. *arXiv preprint arXiv:2104.13326*, 2021.

[ZL15] Yuchen Zhang and Xiao Lin. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 353–361. PMLR, 2015.

[ZZ20] Daoli Zhu and Lei Zhao. Linear convergence of randomized primal-dual coordinate method for large-scale linear constrained convex programming. In *International Conference on Machine Learning*, pages 11619–11628. PMLR, 2020.

[ZZ22] Lei Zhao and Dao-Li Zhu. On iteration complexity of a first-order primal-dual method for nonlinear convex cone programming. *Journal of the Operations Research Society of China*, 10(1):53–87, 2022.