©Copyright 2019

Rahul Kidambi

# Stochastic Gradient Descent For Modern Machine Learning: Theory, Algorithms And Applications

Rahul Kidambi

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

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Sham M. Kakade, Chair

Maryam Fazel

Zaid Harchaoui

Program Authorized to Offer Degree:
Department of Electrical and Computer Engineering

University of Washington

## Abstract

Stochastic Gradient Descent For Modern Machine Learning:
Theory, Algorithms And Applications

Rahul Kidambi

Chair of the Supervisory Committee:
Associate Professor Sham M. Kakade
Paul G. Allen School of Computer Science and Department of Statistics

Tremendous advances in large scale machine learning and deep learning have been powered by the seemingly simple and lightweight stochastic gradient method [99]. Variants [103, 94] of the stochastic gradient method (based on iterate averaging) are known to be asymptotically optimal (in terms of predictive performance). This thesis examines non-asymptotic issues surrounding the use of stochastic gradient descent (SGD) in practice with an aim to achieve its asymptotically optimal statistical properties. Focusing on the stochastic approximation problem of least squares regression, this thesis considers:

1. Understanding the benefits of tail-averaged SGD, and understanding how SGD's non-asymptotic behavior is influenced when faced with mis-specified problem instances.

2. Understand the parallelization properties of SGD, with a specific focus on mini-batching, model averaging and batch size doubling. Can this characterization shed light on algorithmic regimes (for *e.g.* largest instance dependent batchsizes) that admit linear parallelization speedups over vanilla SGD (with a batch size 1), thus presenting useful prescriptions that make best use of our hardware resources whilst not being wasteful of computation? As a byproduct of these results, can we understand how the learning rate behaves as a function of the batch size?

3. Similar to how momentum/acceleration schemes such as heavy ball momentum [92], or Nesterov's acceleration [85] improve over standard batch gradient descent, can we formalize improvements achieved by accelerated methods when working with sampled stochastic gradients? Is there an algorithm that achieves this improvement over SGD? How does deterministic accelerated schemes such as heavy ball momentum, or say, Nesterov's acceleration work when used with sampled stochastic gradients?

4. This thesis considers the behavior of the final iterate of SGD (as opposed to a majority of efforts in the stochastic approximation literature which focus on iterate averaging) with varying stepsize schemes, including the standard polynomially decaying stepsizes [99, 103, 94] and the practically preferred step decay scheme, with an aim to achieve minimax rates. The overarching goal of this section is to understand the behavior of SGD's final iterate owing to its widespread use in practical implementations for machine learning applications.

Alongside the theory results that focus on the least squares regression, this thesis examines the general applicability of various results (in a qualitative sense) towards the problem of training multi-layer deep neural networks on benchmark datasets, and presents several useful implications when training deep learning models of practical interest.

# TABLE OF CONTENTS

# LIST OF FIGURES

# GLOSSARY

MLE: Maximum likelihood estimator (or, estimation).

ERM: Empirical risk minimizer (or, minimization).

SGD: Stochastic gradient descent.

HB: Heavy ball (momentum) method.

NAG: Nesterov's accelerated gradient method.

ASGD: Accelerated stochastic gradient method.

LSR: Least squares regression.

# ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my adviser Sham M. Kakade. Sham has been an amazing source of inspiration with his persistence and drive to get clarity and push the boundaries of our understanding. Grad school has been special in many ways, but, the most important bit was meeting Sham and working with him. I hope that Sham's enthusiasm, creativity and views rub off on my work (and life in general).

I would also like to express my gratitude to my other collaborators, including, Praneeth Netrapalli, Prateek Jain, Rong Ge, Aaron Sidford and Krishna Pillutla. It has been a pleasure working with all of you, and this thesis wouldn't exist if not for each of these collaborations. I would like to specifically thank Praneeth for close collaboration and guidance, both with regards to this thesis and beyond. I would also like to thank my committee − Profs. Zaïd Harchaoui, Maryam Fazel and Aleksandr Aravkin for their feedback on this thesis. I would also like to thank Zaïd for his honest feedback and advise on research and life. I would also like to thank Profs. Sreeram Kannan, Kevin Jamieson, Sewoong Oh for their support and encouragement at various crossroads that I have faced during my PhD. I would also like to express my gratitude to Dr. S. Sundararajan, S. Sathiya Keerthi and Vinod Nair for their mentorship at MSR − my perspective has benefited tremendously from our interactions.

Most of my PhD has been completed whilst hanging out at the AI & ML lab in the Allen and Gates center for CS at UW. I am very grateful to the CS department for giving me access to their wonderful environment, office space, and treat me as one of their own. In particular, I would like to thank Elise Dorough, who has helped me quite a bit during my PhD. I would also like to thank Brenda Larson from ECE, who has helped me navigate various administrative procedures.

I would like to thank my fellow students with whom I have enjoyed sharing offices, views and many collegial moments. This includes: Rob Gens, Tyler Johnson, Marco Ribeiro, Abe Friesen, Nelson Liu, John Thickstun, Colin Lockard, Robbie Weber, Kendall Lowrey, Sam Ainsworth, Ian Covert, Andrew Wagenmaker, Vincent Roulet, Rahul Nadkarni, and more generally, the UW ML group. My friends from CSE, including, Krishna, Srini, Makrand, Akshay, Siva, Guna, Aravind, Aditya have been a band of brothers away from home. Closer to home, my friends Srinivas N.G., Hari K., Ambarisha, Shaishav, Abhijit, Prahadeesh, Rishab, Phani, Sai, Apoorvaa, Pritish, Adharsh, Adrien have been sources of support and happiness.

Finally, my family – firstly, thanks to my extended family (especially my in-laws) for their support and prayers towards completing this thesis. My uncle and his family have been sources of respite and enjoyment – indeed, my uncle Gopi has taken keen interest in my progress and I am grateful to have his support. My sister Vichitra, brother-in-law Adarsh, niece Ananya and nephew Aditya have played an integral part towards helping me take this journey in perspective, especially during the hard times – their love, support, shelter, food and prayers are gratefully appreciated. My wife, Ranjini has held this together as things have swayed, often way too much for comfort. I thank her for her love, affection and support – I look forward to enjoying my life with you. Finally, my grand parents and parents – it is not an overstatement to say that this is **our** moment, as opposed to being mine. You suffered when I struggled, tirelessly prayed for me, celebrated with gusto for each of my successes along the way – I hope this thesis gives you joy and pride. I am driven to share my love, life and be your pillar of support for the time to come.

Funding from National Science Foundation (NSF) and Washington Research Foundation (WRF) is gratefully acknowledged.

## DEDICATION

To my family –

My Parents: Jayashree Rajan and K. Govindarajan,

My Grand Parents: S. Renganayaki and N. Sadagopan.

# Chapter 1
# INTRODUCTION

Stochastic Gradient Descent (SGD) is the workhorse learning algorithm that drives the advancements in modern machine learning and artificial intelligence applications. Developed in the 1950's by the work of [99], SGD and its variants based on iterate averaging [103, 94] have been shown to be (asymptotically) optimal in a precise statistical sense [83, 94, 1, 96], [120, chap. 5,7,8]. The benefits of SGD (in terms of generalization) for modern large scale Machine Learning and the associated benefits stemming from computation-statistics tradeoffs was highlighted in the influential work of [12]. In fact, tremendous advances stemming from modern Deep Learning models, started in part from the breakthrough work of [61] has been driven at its core by variants of SGD. From an overwhelming perspective of modern large scale machine learning and deep learning research, SGD's impact has indeed been immense and central towards the development of superior downstream technology powered by these models. Despite various successes in practice, SGD's understanding in its current forms poses several striking limitations including issues such as (a) slow convergence, (b) excessive manual tuning on every problem with a view to extract best performance out of a given machine learning model, and (c) how best to parallelize SGD and make best use of our burgeoning supply of computing resources without being wasteful in terms of computation. This thesis undertakes a study into the above issues through the following two-pronged approach:

- precise quantitative results that presents an understanding of the behavior of SGD through optimizing the streaming least squares regression objective, and

- quantitative characterization of the above results generalizing to more complicated problems involving training near state-of-the-art modern deep learning models.

Through this approach, this thesis makes progress on resolving the three issues (described above) that plague SGD's ease in deployment (in practice). The specific contributions of this thesis is summarized as follows:

- Understanding the generalization properties of SGD's variants including tail (*i.e.* suffix) averaged SGD, and presenting a non-asymptotic characterization of how SGD's generalization properties is affected by model mis-specification, which, clearly bears much relevance to every SGD implementation in practice.

- Understanding parallelization properties of SGD's variants, and in particular, presenting a precise treatment of three parallelization schemes (which present varying degrees of computation/communication tradeoffs) namely, mini-batching, model averaging and batch size doubling. This characterization sheds light on regimes when these schemes offer linear parallelization speedups, thus enabling best use of our computing infrastructures whilst retaining computational efficiency. A by-product of this result yields sharp problem dependent characterization of the largest batchsize that offers linear parallelization speedups, and tradeoffs involving learning rate versus batchsize.

- Developing Accelerated SGD, a method that presents a strict running time improvement over SGD to obtain minimax rates for the stochastic approximation problem of least squares regression, and thus yielding the *first* running time improvement over the standard Kaczmarz method [58, 115] for solving a system of consistent linear equations when presented with access to one equation at a time. This thesis also sheds light (from both theory and empirical perspectives) on the behavior of heavy ball momentum [92] and Nesterov's acceleration [85] when working with sampled stochastic gradients. This is supplemented with detailed and systematic experiments that ascertains the performance improvements offered by accelerated SGD over heavy ball momentum/Nesterov's acceleration/standard SGD for practically applicable problems involving training the current generation of deep neural networks.

- The final contribution of this thesis involves understanding the behavior of the final iterate (as opposed to using iterate averaging) of SGD (with a view to obtain minimax rates) using both standard polynomially decaying stepsizes [99, 94] as well as the practically widely used step decay learning rate schedule. This result presents reasons that support the use of the stepdecay schedule and presents implications that sheds light on deploying variants of SGD that are amenable towards achieving strong performance with limited manual tuning. Empirical results corroborating with the theoretical contributions indicate performance benefits offered by using the step decay schedule for training modern deep neural networks.

The main contributions of this thesis serves to not only supplant our (theoretical) understanding of SGD in a way that aligns stochastic approximation theory to mirror SGD's behavior in practice; it also follows up with developing learning methods and procedures that are not manually intensive and are strictly easier to deploy in practice for machine learning and deep learning problems at scale.

# Chapter 2
# BACKGROUND

Much of the recent progress in Machine Learning has been set in the context of Supervised Learning. In the setup of Supervised Learning, the learning Algorithm is provided with a "training" set of $n$ input-output pairs denoted as $\mathcal{S}_n \equiv \{\mathbf{x}_i, y_i\}_{i=1}^n$, with $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ being the inputs/features/covariates and $y_i \in \mathcal{Y}$ representing the targets/predictor variables. Typically, it is assumed that the inputs and outputs are drawn from some (unknown) probability distribution $\mathcal{D}$. Now, it is possible that the set $\mathcal{Y}$ could either be:

- Discrete, which is the case with classification problems, where $y_i$ are categorical.
- Continuous, which is typically the case with regression problems, where $y_i$ are continuous real valued.

Supervised Learning involves learning a predictor $g(\cdot)$ that maps inputs $\mathbf{x} \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$. The learning procedure involves searching for the mapping function $g : \mathcal{X} \to \mathcal{Y}$ from the set of all possible functions, which is denoted by $\mathcal{G}$. An example of a predictor function $g(\cdot)$ could be a linear predictor i.e. $g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$, where $\mathbf{w}$ are the model parameters. Another example suited to the modern context is a deep learning model where $g(x) = h_l(\mathbf{w}_l h_{l-1}(\mathbf{w}_{l-1} ... h_1(\mathbf{w}_1 \mathbf{x})...))$ where $h_j$ are non-linearities and $\mathbf{w}_j$ are model parameters.

## 2.1 *Types of Risk: Sample and Population*

The best function $g_{\text{best}}(\cdot) \in \mathcal{G}$ is one that minimizes a criterion (also referred to as risk/ loss/ objective function). The criterion function provides a measure of mismatch between the predictions $g(\mathbf{x})$ and the true values $y$ on the samples $(\mathbf{x}, y) \sim \mathcal{D}$. Two popular examples of the criterion function are:

- The log loss, used when $y_i \in \{-1, 1\}$: $\ell(y, g(\mathbf{x})) = \log(1 + \exp(-yg(\mathbf{x})))$.
- The square loss, as commonly employed in regression with $\ell(y, g(\mathbf{x})) = \frac{1}{2}(y - g(\mathbf{x}))^2$.

Among the types of risk that we can consider optimizing, there exist at least two variants of interest: (i) The training/empirical/in-sample risk and (ii) The testing/population/out of sample risk; these are defined below.

### *2.1.1 Sample/Empirical Risk*

The training risk (or) the training loss of an estimation procedure that returns an estimate $g \in \mathcal{G}$ of the "true" mapping function is defined to be:

$$R_n(g) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, g(\mathbf{x}_i)). \tag{2.1}$$

In particular, this is the loss on the training set $\mathcal{S}_n$ given to the learning algorithm. Note that the training loss is computable exactly since we have access to the training set $\mathcal{S}_n$. Often, we optimize the empirical risk $R_n(g)$ weighted by a regularizer $\Omega(g)$ which promotes obtaining a parsimonious solution (for e.g., measured by a Hilbertian norm).

### *2.1.2 Population Risk*

The testing risk (or) the testing loss of an estimation procedure that returns an estimate $g \in \mathcal{G}$ of the "true" mapping function is:

$$R(g) \overset{\text{def}}{=} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left[\ell(y, g(\mathbf{x}))\right]. \tag{2.2}$$

All of Machine Learning revolves around the quality of the predictions offered by a mapping function on an unknown instance $(\mathbf{x}, y) \sim \mathcal{D}$ that is not seen as a part of the learning process; this emphasizes the importance of controlling and minimizing the testing error. Note that unlike the training error, the test error is not exactly computable since we do not have access to the distribution $\mathcal{D}$ from which the data is drawn.

## 2.2 Linear Least Squares Regression

In this thesis, all our (theoretical) discussions consider the case where $\hat{y} = g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$, and the square loss case, i.e., $\ell(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(y - \mathbf{w}^\top \mathbf{x})^2$. In this case, we express the empirical risk for a set of samples $\mathcal{S}_n \equiv \{\mathbf{x}_i, y_i\}_{i=1}^n$ in the following manner:

$$R_n(\mathbf{w}) = \frac{1}{2n} \sum_{i=1}^n (y_i - \mathbf{w}^\top \mathbf{x}_i)^2. \tag{2.3}$$

In a similar manner, we can express the population risk as:

$$R(\mathbf{w}) = \frac{1}{2} \cdot \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left[ (y - \mathbf{w}^\top \mathbf{x})^2 \right]. \tag{2.4}$$

## 2.3 Estimation and Approximation Error

Considering the objectives written out in equations (2.3), (2.4), we will define estimation and approximation error, and for this purpose, we require the definition of a minimizer of the empirical and population risk. For starters, let us begin by defining the Hessian $\mathbf{H}$ of the population risk in equation (2.4):

$$\mathbf{H} \overset{\text{def}}{=} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_x} \left[ \mathbf{x}\mathbf{x}^\top \right]. \tag{2.5}$$

Let us assume that the Hessian $\mathbf{H}$ is invertible, i.e., $\mathbf{H} \succ 0$ (where, $\succ$ represents the Lowener ordering). This implies the objective in equation (2.4) is strongly convex and admits a unique minimizer $\mathbf{w}^*$, i.e.,

$$\mathbf{w}^* \overset{\text{def}}{=} \arg\min_{\mathbf{w}\in\mathbb{R}^d} R(\mathbf{w}). \tag{2.6}$$

A crucial quantity of interest in characterizing the underlying estimation problem is the noise covariance matrix $\mathbf{\Sigma}$, and this is expressed as:

$$\mathbf{\Sigma} \overset{\text{def}}{=} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left[ (y - \langle \mathbf{w}^*, \mathbf{x} \rangle)^2 \mathbf{x}\mathbf{x}^\top \right]. \tag{2.7}$$

In a similar manner, considering the empirical risk in equation (2.3), denote the empirical Hessian to be $\mathbf{H}_n$ and the empirical risk minimizer (ERM) over samples $\mathbf{S}_n$ to be $\mathbf{w}_n^*$:

$$\mathbf{H}_n \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \tag{2.8}$$

$$\mathbf{w}^*_{\text{ERM}} \overset{\text{def}}{=} \arg\min_{\mathbf{w}\in\mathbb{R}^d} R_n(\mathbf{w}). \tag{2.9}$$

Let $\widehat{\mathbf{w}}$ be a hypothesis returned through an estimation procedure. The excess risk of this hypothesis is defined as:

$$R(\widehat{\mathbf{w}}) - R(\mathbf{w}^*) = \underbrace{R(\widehat{\mathbf{w}}) - R(\mathbf{w}^*_n)}_{\text{Optimization Error}} + \underbrace{R(\mathbf{w}^*_n) - R(\mathbf{w}^*)}_{\text{Estimation Error}}. \tag{2.10}$$

The estimation error captures several facets of the underlying learning problem, and this includes issues such as (a) *noise:* Consider the case when the model is well specified (also referred to as the additive noise case). In this situation, the empirical risk minimizer has a risk that is larger than the population risk minimizer by an amount that is related to the variance of the distribution over the output $y$ conditioned on a given input $\mathbf{x}$. For example, in the case of linear least squares regression, ERM can be thought of as maximum likelihood estimation (MLE) under a Gaussian noise model. In particular, solving for the ERM is equivalent to the statistical model $y \sim \mathcal{N}(\langle \mathbf{w}^*, \mathbf{x}\rangle, \sigma^2)$ for samples $(\mathbf{x}, y) \sim \mathcal{D}$. (b) *model mismatch:* This is the situation when the model is not well specified (also referred to as the non-realizable case). More specifically, this is the case when the data is generated from a parametric model class that is different from the one that is being optimized over. In the context of least squares, a characteristic of model mismatch is when there exists dependencies between the error $y - \langle \mathbf{w}^*, \mathbf{x}\rangle$ and the input $\mathbf{x}$. The optimization error can be viewed as error stemming from not computing the exact optimizer of the (possibly) empirical risk given samples $\mathcal{S}_n$ using the learning algorithm.

### *2.3.1 Minimax Rates of Estimation*

At the heart of the parameter estimation problem underlying supervised learning (which can be interpreted as Maximum Likelihood Estimation with an appropriate noise model), it is crucial to understand a lower bound on the approximation error for any learning algorithm with access to $n$ samples from the distribution $\mathcal{D}$. Given implications of the approximation error on the generalization error bound and the impact of approximation error on issues such as model mismatch and the variance inherent even in a well specified model, a lower bound is very valuable towards selecting a learning algorithm suited to the problem at hand. Before going over this (minimax) lower bound, we require the definition of the noise variance $\widehat{\sigma^2_{\text{MLE}}}$:

$$\widehat{\sigma^2_{\text{MLE}}} \stackrel{\text{def}}{=} \operatorname{Tr}\left(\mathbf{H}^{-1}\mathbf{\Sigma}\right). \tag{2.11}$$

In the case of well specified models, $(y - \langle \mathbf{w}^*, \mathbf{x}\rangle)$ is independent of $\mathbf{x}$, thus

$$\mathbf{\Sigma} = \mathbb{E}\left[(y - \langle \mathbf{w}^*, \mathbf{x}\rangle)^2 \mathbf{x}\mathbf{x}^\top\right] = \sigma^2\mathbf{H},$$

where, $\sigma^2 = \mathbb{E}\left[(y - \langle \mathbf{w}^*, \mathbf{x}\rangle)^2\right]$. This implies that for the special case of well specified models, $\widehat{\sigma^2_{\text{MLE}}} = d\sigma^2$. Given $n$ samples from the distribution $\mathcal{D}$ this minimax statistical rate is $\widehat{\sigma^2_{\text{MLE}}}/n$ [4, 33] [64, chap. 5,10,11]. This minimax rate sets a baseline for any learning algorithm to compete in: in particular, regardless of the rates offered by any optimization algorithm on say the empirical risk (equation (2.3)), the rate on the generalization error can never be improved beyond this minimax rate.

It is known that under certain regularity conditions employed in Statistics literature, given $n$ samples $\mathcal{S}_n$ from the distribution $\mathcal{D}$, the Maximum Likelihood Estimator (or the Empirical Risk Minimizer) achieves this statistical minimax rate, in the limit of large $n$ [62, chap. 1,3,6], [120, chap. 5,7,8], [69, chap. $5-6$], i.e.:

$$\lim_{n\to\infty} \frac{\mathbb{E}_{\mathcal{S}_n}\left[R(\mathbf{w}^*_{\text{ERM}})\right] - R(\mathbf{w}^*)}{\widehat{\sigma^2_{\text{MLE}}}/n} = 1, \tag{2.12}$$

where, the expectation is over all possible $n$ samples $\mathcal{S}_n$ drawn from the distribution $\mathcal{D}$. It is important to note that this result is a rather general result as it holds for a broad class of models in the agnostic setting (i.e. as in the case of model mismatch).

In several situations, the ERM (computed upto machine precision) satisfies equation (2.12) and hence it is a natural idea to optimize for the empirical risk represented by equation (2.3) since it possesses favourable statistical properties. This motivates the study of offline optimization, which aims to optimize the function in equation (2.3).

## *2.4 Offline optimization*

In the study of minimization methods for problem (2.3), first order methods, which access a function to be optimized using its gradient information has formed the cornerstone of modern machine learning and optimization. In this section, we review convex optimization methods that employ an exact first order oracle [83, 86]. The basic and the most popular method in this category is Gradient Descent [18]. Gradient Descent makes progress on the objective (2.3) by following the direction of negative gradient, i.e.:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta \nabla R_n(\mathbf{w}_{t-1}).$$

with $\eta > 0$ being the learning rate. Note that the computation of the full exact (batch) gradient $\nabla R_n(\mathbf{w}_{t-1})$ requires $\mathcal{O}(nd)$ computation. Now, to understand the computational effort needed to attain an $\epsilon$-accurate solution, we begin by defining the condition number $\kappa_o$ of the optimization problem in equation (2.3).

$$\kappa_o = \frac{\lambda_{\max}(\mathbf{H}_n)}{\lambda_{\min}(\mathbf{H}_n)}.$$

where, $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalues of the Hessian $\mathbf{H}_n$ (introduced in equation (2.8)). Under an appropriate choice of the learning rate, $\kappa_o \log(1/\epsilon)$ iterations of Gradient Descent are required to achieve $\epsilon$-accuracy to the minimization problem. This is known to be suboptimal amongst the class of methods with access to a first order oracle [86]. The total computational effort of Gradient Descent to achieve $\epsilon$-accuracy is $\mathcal{O}(nd\kappa_o \log(1/\epsilon))$.

The suboptimal behavior of Gradient Descent has been addressed through the development of fast gradient methods such as Conjugate Gradient Method [48] (for a linear system), the Heavy Ball method [92] (for quadratic objectives), the Accelerated Gradient Descent Method [85] (general smooth convex functions). These methods reduce the $\kappa_o$ to $\sqrt{\kappa_o}$ iterations, i.e., they require $\mathcal{O}(\sqrt{\kappa_o} \log(1/\epsilon))$ iterations to achieve an $\epsilon$-accurate solution, which is known to be optimal amongst the class of methods with access to an exact first order oracle [86]. This implies fast gradient methods have a computational effort of $\mathcal{O}(nd\sqrt{\kappa_o} \log(1/\epsilon))$ to achieve $\epsilon$-accuracy.

The multiplicative coupling of the condition number $\kappa_o$ (or $\sqrt{\kappa_o}$) and the number of data points $n$ in the complexity bounds for Gradient Descent and its accelerated variants renders it unsuitable for modern large scale machine learning. In particular, modern large scale machine learning has seen an upsurge in the number of data points $n$ and the condition number for several practical problems could be as large as $\mathcal{O}(n)$; a multiplicative coupling of these quantities leads to super linear running time in the number of data points $n$, which is often infeasible for modern large scale problems.

A recent upsurge of offline stochastic methods has seen the development of several variants of the sub-linearly convergent Stochastic Gradient Descent (SGD) [99, 97]. These variants offer linearly convergent algorithms that allow for the coupling between $n$ and $\kappa_o$ to be additive as opposed to multiplicative; in essence, this leads to nearly linear (in $n$) running time for these methods. Amongst these are SDCA [108], SAG [101], SVRG [57], SAGA [24], and these have running time guarantees that resemble $\mathcal{O}((n+\bar{\kappa_o})\cdot d\log(1/\epsilon))$, with $\bar{\kappa_o}$ representing a dependence on the trace of $\mathbf{H}_n$ as opposed to its spectral norm. These methods have been followed up with their accelerated variants such as [109, 34, 72, 23, 2] and these accelerated variants possess improved running times of $\mathcal{O}((n+\sqrt{n\bar{\kappa_o}})\cdot d\log(1/\epsilon))$[1]. These running times have been shown to be nearly optimal through the work of [124].

## 2.5 Stochastic Approximation

The seminal work of [99] introduced the problem of stochastic approximation (which is expressed as a variant of the problem expressed in equation (2.2) [2]) and proposed a method to solve the stochastic approximation problem. The method they introduced to solve the stochastic approximation problem is popularly termed as Stochastic Gradient Descent (SGD), which moves from an iterate $\mathbf{w}_k$ to $\mathbf{w}_{k+1}$ using an unbiased estimate of the gradient $\widehat{\nabla R}(\mathbf{w})$

[1]Accelerated variants thus indicate improvements only in situations when $n \leq \bar{\kappa_o}$

[2][99] motivated the problem as searching for the root of a function whose evaluations can be carried out in an unbiased but inexact manner. Such a view can be employed for the problem in equation (2.2) where we search for an iterate $\mathbf{w}$ where the gradient of the population risk vanishes, i.e., $\nabla R(\mathbf{w}) = 0$.

evaluated at $\mathbf{w} = \mathbf{w}_k$, i.e.,

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \gamma_k \widehat{\nabla R}(\mathbf{w}_k), \text{ where, } \widehat{\nabla R}(\mathbf{w}_k) = \nabla R(\mathbf{w}_k) + \epsilon_{k+1}$$

with $\mathbb{E}\left[\epsilon_{k+1}|\mathcal{F}_k\right] = 0$, $\mathcal{F}_k$ being a filtration formed using events leading to $\mathbf{w}_k$. $\{\gamma_k\}_{k\geq 1}$ is a sequence of decreasing step sizes satisfying certain properties. A breakthrough in the context of stochastic approximation appeared through the works of [103, 94] and this was through understanding the properties of a certain iterate averaged version of SGD. They proved that under certain regularity conditions, in an asymptotic sense, iterate averaged SGD converges in distribution to the MLE. What this implies in particular is that the excess risk of the iterate averaged SGD estimator achieves the minimax optimal statistical risk that is satisfied by the MLE (equation (2.12)) [62, chap. 1,3,6], [69, chap. $5-6$], [120, chap. 5,7,8]. Furthermore, iterate averaged SGD [103, 94] allowed for the use of a larger step size sequence $\{\gamma_k\}_{k\geq 1}$ compared to the results of [99].

Before moving to issues relating to non-asymptotics of stochastic approximation, we will make a digression here with regards to highlighting other notions of optimality studied in the literature of adaptive signal processing/control theory with regards to the behavior of stochastic gradient descent (which is referred to as the least mean square (LMS) algorithm [122]). In particular, it is known that the LMS algorithm is robust (and optimal) in a $H^\infty$ sense [44], i.e., small disturbances (*i.e.* noise) and modeling errors (defined in an appropriate sense) provably leads to small estimation errors.

In order to progress towards developing a non-asymptotic understanding of Stochastic Approximation, we require grounding the information that we can access in the context of an oracle model, in a similar manner to how we understood the behavior of (offline) Gradient Descent and other fast Gradient methods. For starters, let us consider the least squares population risk expressed in equation (2.4). In the context of Stochastic Approximation, we are provided with an access to a *stochastic first order oracle*, which when queried at any iterate $\mathbf{w}$, samples a tuple $(\mathbf{x}, y) \sim \mathcal{D}$ and returns an unbiased estimate of the gradient of the objective (2.4) using this sample:

$$\widehat{\nabla R}(\mathbf{w}) = -(y - \langle \mathbf{w}, \mathbf{x} \rangle) \cdot \mathbf{x};\ \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\widehat{\nabla R}(\mathbf{w})\right] = \nabla R(\mathbf{w}). \tag{2.13}$$

This oracle model represents the information that is accessed by practical implementations of stochastic gradient descent. Differences between this oracle model and other theoretically useful first order stochastic oracles [83, 86] is detailed in subsequent sections.

As a final note, the excess risk of an estimation procedure (say, SGD) used for the Stochastic Approximation problem is composed of two components [5, 7, 35]: (i) the *bias error*, which relates to the dependence of the risk on the initial conditions (for example, the excess risk of the starting iterate $\mathbf{w}_0$). The bias error typically dominates the non-asymptotic behavior of SGD, and (ii) the *variance error*, which relates to the error introduced owing to the use of the stochastic first order oracle. The variance error is typically leading order and associated with statistical/information theoretic lower bounds [83, 1].

## 2.6 Stochastic Approximation and its suitability to Large Scale Learning

As mentioned before, minimizing the test error (or) the generalization error is key towards the success of Machine Learning. One wave of Machine Learning considered the Empirical Risk Minimization framework (say, equation (2.1), or in particular, equation (2.3)) as a gold standard to progress towards minimization of the generalization error, owing to the favorable statistical properties of the MLE (equation (2.12)). However, an exact solution, say to the problem in equation (2.3) requires a computational cost of $\mathcal{O}(nd^2 + d^\omega)$, where $\omega$ is the matrix multiplication exponent, and $\mathcal{O}(d^2)$ storage thus being impermissible for large $d$. In general, exact (machine precision or like) solutions to the ERM problem tends to require super linear dependence on computation or storage (for example, to solve the Newton step once or multiple times) making them infeasible for modern problems.

Granted, a wave of recent progress in the large scale offline optimization literature (as detailed in section 2.4) has allowed for the development of algorithms with near linear running times in the number of data points $n$. However, it is unclear how to connect these developments on achieving $\epsilon$-accurate solutions to the ERM to generalization without resorting to upper bounds. Note that any rate proven in terms of the empirical risk (equation (2.3)) does not naturally translate to rates on the population risk (equation (2.4)). In particular, for such a translation to occur, there need to be additional arguments regarding concentration of the empirical risk, a new generalization error analysis translating the training error to the test error. Moreover, it is not clear how the condition numbers of problem in equation (2.4) and equation (2.3) are related. A final, but crucial aspect of methods described in section 2.4 is that they require for the entire dataset to be stored in memory, and for multiple passes (in particular, $\log n$ passes) to be taken over this dataset stored in memory. Refer to [35] for more details.

While these issues with the ERM framework seem rather fundamental, it is fortunate that the aim in Machine Learning is in minimizing the generalization error. This allows us to consider approaches that optimize directly this quantity of interest. Knowing the fact that there exists fundamental minimax lower bounds on the generalization error [4, 33] [64, chap. 5,10,11], and given the issues (highlighted above) with the ERM framework, it is prudent to focus more on achieving this statistical accuracy on the testing risk as opposed to solving the ERM problem to numerical precision. This clearly indicates the relevance of the stochastic approximation framework that aims to optimize the quantity of interest, say, in equation (2.2) or equation (2.4). Moreover, algorithms such as SGD are single pass streaming methods and do not require storing an entire dataset in memory, which is well suited to large scale machine learning problems. These observations and the relevant issues of computation-statistics tradeoffs was highlighted in the work of [12].

# Chapter 3

# TAIL-AVERAGED SGD: PARALLELIZATION VIA MINI-BATCHING, MODEL AVERAGING AND BATCHSIZE DOUBLING

## *3.1 Chapter Notes*

This chapter presents joint work with Prateek Jain, Praneeth Netrapalli, Sham M. Kakade and Aaron Sidford and is published in the Journal of Machine Learning Research (JMLR), 2018. This chapter's presentation is a (very minor) variant of the published version of this paper. The contributions of this chapter are summarized as follows:

- This chapter establishes the benefits of tail-averaging (also known as suffix averaging) using the bias-variance decomposition in the context of stochastic approximation, through an illustrative instance of streaming strongly convex least squares regression.

- This chapter presents a precise characterization of the advantages of mini-batch SGD, in terms of the parallelization benefits offered over SGD with a batch size 1, both when the bias is the dominating source of error (at the start of learning, when the error sharply drops) and when the variance is the dominating source of error (when the error bounces around). A byproduct of this analysis indicates how the learning rate must be set as a function of the batchsize, presenting guidelines to practitioners when using mini-batching (which is the most ubiquitously used form of parallelization). The chapter also presents a precise view into how mini-batching works with and without iterate averaging.

- This chapter establishes a precise characterization of model averaging, which involves running several instances of SGD in separate machines, and averaging the results of these runs once (at the end of optimization); such procedures are meaningful alternatives in situations when communication between various processors is deemed very expensive.

- This chapter then considers the role of model mis-specification and when considering linear regression problem instances where the noise in each example need not be bounded. Such results presents a clear insight into the generalization behavior of SGD in these (realistic) settings, and results in this direction are original to the best of our knowledge.

- A final contribution of this chapter is in developing an SGD based batchsize doubling approach that offers the same generalization error (after nearly the same number of serial updates) as batch gradient descent, thus presenting a highly parallelizable streaming algorithm that competes with batch gradient descent on every least squares problem instance.

## *3.2 Introduction and Problem Setup*

With the ever increasing size of modern day datasets, practical algorithms for machine learning are increasingly constrained to spend less time and use less memory. This makes it particularly desirable to employ simple streaming algorithms that generalize well in a few passes over the dataset.

Stochastic gradient descent (SGD) is perhaps the simplest and most well studied algorithm that meets these constraints. The algorithm repeatedly samples an instance from the stream of data and updates the current parameter estimate using the gradient of the sampled instance. Despite its simplicity, SGD has been immensely successful and is the de-facto method for large scale learning problems. The merits of SGD for large scale learning and the associated computation versus statistics tradeoffs is discussed in detail by the seminal work of [12].

While a powerful machine learning tool, SGD in its simplest forms is inherently serial. Over the past years, as dataset sizes have grown there have been remarkable developments in processing capabilities with multi-core/distributed/GPU computing infrastructure available in abundance. The presence of this computing power has triggered the development of parallel/distributed machine learning algorithms ([76, 130, 15, 90, 71, 129]) that possess the capability to utilize multiple cores/machines. However, despite this exciting line of work, it is yet unclear how to best parallelize SGD and fully utilize these computing infrastructures.

This chapter takes a step towards answering this question, by characterizing the behavior of constant stepsize SGD for the problem of strongly convex stochastic least square regression (LSR) under two averaging schemes widely believed to improve the performance of SGD. In particular, this chapter considers the natural parallelization technique of *mini-batching*, where multiple data-points are processed simultaneously and the current iterate is updated by the average gradient over these samples, and combine it with variance reducing technique of *tail-averaging*, where the average of many of the final iterates are returned as SGD's estimate of the solution.

In this chapter, parallelization arguments are structured through the lens of a *work-depth* tradeoff: *work* refers to the total computation required to reach a certain generalization error, and *depth* refers to the number of serial updates. Depth, defined in this manner, is a reasonable estimate of the runtime of the algorithm on a large multi-core architecture with shared memory, where there is no communication overhead, and has strong implications for parallelizability on other architectures.

### *3.2.1 Problem Setup and Notations*

We use boldface small letters ($\mathbf{x}, \mathbf{w}$ etc.) for vectors, boldface capital letters ($\mathbf{A}, \mathbf{H}$ etc.) for matrices and normal script font letters ($\mathcal{M}$,$\mathcal{U}$ etc) for tensors. We use $\otimes$ to denote the outer product of two vectors or matrices. Loewner ordering between two PSD matrices is represented using $\succeq, \preceq$.

This chapter considers the stochastic approximation problem of Least Squares Regression

(LSR). Let $L : \mathbb{R}^d \to \mathbb{R}$ be the expected square loss over tuples $(\mathbf{x}, y)$ sampled from a distribution $\mathcal{D}$:

$$L(\mathbf{w}) = \frac{1}{2} \cdot \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[(y - \langle \mathbf{w}, \mathbf{x} \rangle)^2] \; \forall \; \mathbf{w} \in \mathbb{R}^d. \tag{3.1}$$

Let $\mathbf{w}^*$ be a minimizer of the problem (3.1). Now, let the Hessian of the problem (3.1) be denoted as:

$$\mathbf{H} \overset{\text{def}}{=} \nabla^2 L(\mathbf{w}) = \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right].$$

Next, we define the fourth moment tensor $\mathcal{M}$ of the inputs $\mathbf{x}$ as:

$$\mathcal{M} \overset{\text{def}}{=} \mathbb{E}\left[\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}\right].$$

Let the noise $\epsilon_{\mathbf{x},y}$ in a sample $(\mathbf{x}, y) \sim \mathcal{D}$ with respect to the minimizer $\mathbf{w}^*$ of (3.1) be denoted as:

$$\epsilon_{\mathbf{x},y} \overset{\text{def}}{=} y - \langle \mathbf{w}^*, \mathbf{x} \rangle.$$

Finally, let the noise covariance matrix $\mathbf{\Sigma}$ be denoted as:

$$\mathbf{\Sigma} \overset{\text{def}}{=} \mathbb{E}\left[\epsilon^2_{\mathbf{x},y}\mathbf{x}\mathbf{x}^\top\right].$$

The *homoscedastic* (or, additive noise/well specified) case of LSR refers to the case when $\epsilon_{\mathbf{x},y}$ is mutually independent from $\mathbf{x}$. This is the case, say, when $\epsilon_{\mathbf{x},y}$ sampled from a Gaussian, $N(0, \sigma^2)$ independent of $\mathbf{x}$. In this case, $\mathbf{\Sigma} = \sigma^2\mathbf{H}$, where, $\sigma^2 = \mathbb{E}\left[\epsilon^2\right]$, where the subscript on $\epsilon_{\mathbf{x},y}$ is suppressed owing to the independence of $\epsilon$ on any sample $(\mathbf{x}, y) \sim \mathcal{D}$. On the other hand, the *heteroscedastic* (or, mis-specified) case refers to the setting when $\epsilon_{\mathbf{x},y}$ is correlated with the input $\mathbf{x}$. In this chapter, all our results apply to the general mis-specified case of the LSR problem.

### *Assumptions*

We make the following assumptions about the problem.

($\mathcal{A}1$) **Finite fourth moment:** The fourth moment tensor $\mathcal{M} = \mathbb{E}\left[\mathbf{x}^{\otimes 4}\right]$ exists and is finite.

($\mathcal{A}2$) **Strong convexity:** The Hessian of $L(\cdot)$, $\mathbf{H} = \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right]$ is positive definite i.e., $\mathbf{H} \succ 0$.

($\mathcal{A}1$) is a standard regularity assumption for the analysis of SGD and related algorithms. ($\mathcal{A}2$) is also a standard assumption and guarantees that the minimizer of (3.1), i.e., $\mathbf{w}^*$ is unique.

*Important Quantities*

In this section, we will introduce some important quantities required to present our results. Let $\mathbf{I}$ denote the $d \times d$ identity matrix. For any matrix $\mathbf{A}$, $\mathcal{M}\mathbf{A} \overset{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{x}^\top \mathbf{A}\mathbf{x}\right)\mathbf{x}\mathbf{x}^\top\right]$. Let $\mathcal{H}_{\mathcal{L}} = \mathbf{H} \otimes \mathbf{I}$ and $\mathcal{H}_{\mathcal{R}} = \mathbf{I} \otimes \mathbf{H}$ represent the left and right multiplication operators of the matrix $\mathbf{H}$ so that for any matrix $\mathbf{A}$, we have $\mathcal{H}_{\mathcal{L}}\mathbf{A} = \mathbf{H}\mathbf{A}$ and $\mathcal{H}_{\mathcal{R}}\mathbf{A} = \mathbf{A}\mathbf{H}$.

- **Fourth moment bound:** Let $R^2$ be the smallest number such that $\mathcal{M}\mathbf{I} \preceq R^2\mathbf{H}$.

- **Smallest eigenvalue:** Let $\mu$ be the smallest eigenvalue of $\mathbf{H}$ i.e., $\mathbf{H} \succeq \mu\mathbf{I}$.

The fourth moment bound implies that $\mathbb{E}\left[\|\mathbf{x}\|^2\right] \leq R^2$. Further more, ($\mathcal{A}2$) implies that the smallest eigenvalue $\mu$ of $\mathbf{H}$ is strictly greater than zero ($\mu > 0$).

*Stochastic Gradient Descent: Mini-Batching and Iterate Averaging*

In this chapter, we work with a stochastic first order oracle. This oracle, when queried at $\mathbf{w}$ samples an instance $(\mathbf{x}, y) \sim \mathcal{D}$ and uses this to return an unbiased estimate of the gradient of $L(\mathbf{w})$:

$$\widehat{\nabla L}(\mathbf{w}) = -(y - \langle \mathbf{w}, \mathbf{x}\rangle) \cdot \mathbf{x}; \;\; \mathbb{E}\left[\widehat{\nabla L}(\mathbf{w})\right] = \nabla L(\mathbf{w}).$$

We consider the stochastic gradient descent (SGD) method [99], which minimizes $L(\mathbf{w})$ by following the direction opposite to this noisy stochastic gradient estimate, i.e.:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \gamma \cdot \widehat{\nabla L_t}(\mathbf{w}_{t-1}), \text{ with, } \widehat{\nabla L_t}(\mathbf{w}_{t-1}) = -(y_t - \langle \mathbf{w}_{t-1}, \mathbf{x}_t\rangle) \cdot \mathbf{x}_t$$

with $\gamma > 0$ being a constant step size/learning rate; $\widehat{\nabla L_t}(\mathbf{w}_{t-1})$ is the stochastic gradient evaluated using the sample $(\mathbf{x}_t, y_t) \sim \mathcal{D}$ at $\mathbf{w}_{t-1}$. We consider two algorithmic primitives used in conjunction with SGD namely, mini-batching and tail-averaging (also referred to as iterate/suffix averaging).

Mini-batching involves querying the gradient oracle several times and using the average of the returned stochastic gradients to take a single step. That is,

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \gamma \cdot \left(\frac{1}{b}\sum_{i=1}^{b} \widehat{\nabla L_{t,i}}(\mathbf{w}_{t-1})\right),$$

where, $b$ is the batch size. Note that at iteration $t$, mini-batching involves repeatedly querying the stochastic gradient oracle at $\mathbf{w}_{t-1}$ for a total of $b$ times. For every query $i = 1, ..., b$ at iteration $t$, the oracle samples an instance $\{\mathbf{x}_{ti}, y_{ti}\}$ and returns a stochastic gradient estimate $\widehat{\nabla L_{t,i}}(\mathbf{w}_{t-1})$. These estimates $\{\widehat{\nabla L_{t,i}}(\mathbf{w}_{t-1})\}_{i=1}^{b}$ are averaged and then used to perform a single step from $\mathbf{w}_{t-1}$ to $\mathbf{w}_t$. Mini-batching enables the possibility of parallelization owing to the use of cheap matrix-vector multiplication for computing stochastic gradient estimates. Furthermore, mini-batching allows for the possible reduction of variance owing to the effect of averaging several stochastic gradient estimates.

Tail-averaging (or suffix averaging) refers to returning the average of the final few iterates of a stochastic gradient method as a means to improve its variance properties [103, 94]. In particular, assuming the stochastic gradient method is run for $n-$steps, tail-averaging (or suffix averaging [97]) involves returning

$$\bar{\mathbf{w}} = \frac{1}{n-s}\sum_{t=s+1}^{n} \mathbf{w}_t$$

as an estimate of $\mathbf{w}^*$. Note that $s$ could be thought of as being $cn$, with $c < 1$ being some constant.

Typical excess risk bounds (or, generalization error bounds) for the stochastic approximation problem involve the contribution of two error terms namely, (i) the bias, which refers to the dependence on the starting conditions $\mathbf{w}_0$/initial excess risk $L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and, (ii) the variance, which refers to the dependence on the noise introduced by the use of a stochastic first order oracle.

*Optimal Error Rates for the Stochastic Approximation problem*

Under standard regularity conditions often employed in the statistics literature, the minimax optimal rate on the excess risk is achieved by the standard Empirical Risk Minimizer (or, Maximum Likelihood Estimator) [69, chap. $5-6$], [120, chap. 5,7,8]. Given $n$ i.i.d. samples $\mathcal{S}_n = \{\mathbf{x}_i, y_i\}_{i=1}^n$ drawn from $\mathcal{D}$, define the empirical risk minimization problem as obtaining

$$\mathbf{w}_n^* = \arg\min_{\mathbf{w}} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \langle \mathbf{w}, \mathbf{x}_i\rangle)^2.$$

Let us define the noise variance $\widehat{\sigma^2_{\text{MLE}}}$ to represent

$$\widehat{\sigma^2_{\text{MLE}}} = \mathbb{E}\left[\|\widehat{\nabla L}(\mathbf{w}^*)\|^2_{\mathbf{H}^{-1}}\right] = \text{Tr}[\mathbf{H}^{-1}\boldsymbol{\Sigma}].$$

The asymptotic minimax rate of the Empirical Risk Minimizer $\mathbf{w}_n^*$ on *every problem instance* is $\widehat{\sigma^2_{\text{MLE}}}/n$ [69, 120], i.e.,

$$\lim_{n\to\infty} \frac{\mathbb{E}_{\mathcal{S}_n}[L(\mathbf{w}_n^*)] - L(\mathbf{w}^*)}{\widehat{\sigma^2_{\text{MLE}}}/n} = 1.$$

For the well-specified case (i.e., the additive noise case, where, $\boldsymbol{\Sigma} = \sigma^2\mathbf{H}$), we have $\widehat{\sigma^2_{\text{MLE}}} = d\sigma^2$. Seminal works of [103, 94] prove that tail-averaged SGD, with averaging from start, achieves the minimax rate for the *well-specified* case in the limit of $n \to \infty$.

*Goal:* In this chapter, we seek to provide a non-asymptotic understanding of (a) mini-batching and issues of learning rate versus batch-size, (b) tail-averaging, (c) the effect of the model mis-specification, (d) a batch size doubling scheme for parallelizing statistical estimation, (e) a communication efficient parallelization scheme namely, parameter-mixing/model averaging and (f) the behavior of learning rate versus batch size on the final iterate of the mini-batch SGD procedure, on the behavior of excess risk of SGD (in terms of both the bias and the variance terms) for the streaming LSR problem, with the goal of achieving the minimax rate on every problem instance.

### *3.2.2 This Chapter's Contributions*

The main contributions of this chapter are as follows:

- This chapter shows that mini-batching yields near-linear parallelization speedups over the standard serial SGD (i.e. with batch size 1), as long as the mini-batch size is smaller than a problem dependent quantity (which we denote by $b_{\text{thresh}}$). When batch-sizes increase beyond $b_{\text{thresh}}$, mini-batching is inefficient (owing to the lack of serial updates), thus obtaining only sub-linear speedups over mini-batching with a batch size $b_{\text{thresh}}$. A by-product of this analysis sheds light on how the step sizes naturally interpolate from ones used by standard serial SGD (with batch size 1) to ones used by batch gradient descent.

- While the final iterate of SGD decays the bias at a geometric rate but does not obtain minimax rates on the variance, the averaged iterate [94, 25] decays the bias at a sublinear rate while achieving minimax rates on the variance. This chapter rigorously shows that tail-averaging obtains the best of both worlds: decaying the bias at a geometric rate and obtaining near-minimax rates (up to constants) on the variance. This result corroborates with empirical findings [79] that indicate the benefits of tail-averaging in general contexts such as training Long-Short term memory models (LSTMs).

- Next, this chapter precisely characterizes the tradeoffs of learning rate versus batch size and its effect on the excess risk of the final iterate of an SGD procedure, which provides theoretical evidence to empirical observations [40, 114] described in the context of deep learning and non-convex optimization.

- Combining the above results, this chapter provides a mini-batching and tail-averaging version of SGD that is highly parallelizable: the number of serial steps (which is a proxy for the un-parallelizable time) of this algorithm nearly matches that of *offline gradient descent* and is lower than the serial time of all existing streaming LSR algorithms.

See Table 3.1 for comparison. We note that these results are obtained by providing a tight finite-sample analysis of the effects of mini-batching and tail-averaging with large constant learning rate schemes.

- This chapter provides a non-asymptotic analysis of parameter mixing/model averaging schemes for the streaming LSR problem. Model averaging schemes are an attractive proposition for distributed learning owing to their communication efficient nature, and they are particularly effective in the regime when the estimation error (i.e. variance) is the dominating term in the excess risk. Here, we characterize the excess risk (in terms of both the bias and variance) of the model averaging procedure which sheds light on situations when it is an effective parallelization scheme (in that when this scheme yields linear parallelization speedups).

- All the results in this chapter are established for the *general mis-specified* case of the streaming LSR problem. This establishes a fundamental difference in the behavior of SGD when dealing with mis-specified models in contrast to existing analyses that deal with the bounded mis-specified noise case. In particular, this analysis reveals a surprising insight that the maximal stepsizes (that ensure minimax optimal rates) are a function of the noise properties of the mis-specified problem instance. The main takeaway of this analysis is that the maximal step sizes (that permit achieving minimax rates) for the mis-specified case can be *much lower* than ones employed in the well-specified case: indeed, a problem instance that yields such a separation between the maximal learning rates for the well specified and the mis-specified case is presented.

The tool employed in obtaining these results generalizes the operator view of averaged SGD with batch size 1 [25] and a clear exposition of the bias-variance decomposition from [53] to obtain a sharp bound on the excess risk for mini-batch, tail-averaged constant step-size SGD. Note that the work of [25] does not establish minimax rates while working with large constant step sizes; this shortcoming is remedied by this chapter through a novel sharp analysis that rigorously establishes minimax optimal rates while working with large constant step sizes. Furthermore, note that while straightforward operator norm bounds of the matrix operators suffice to show convergence of the SGD method, they turn out to be pretty loose bounds (particularly for bounding the variance). To tighten these bounds, this chapter presents a fine grained analysis that bounds the trace of the SGD operators when applied to the relevant matrices. The bounds of this chapter and its advantages compared to existing algorithms is indicated in table 3.1.

While this chapter's results focus on strongly convex streaming least square regression, we believe that our techniques and results extend more broadly. This chapter aims to serve as the basis for future work on analyzing SGD and parallelization of large scale algorithms for machine learning.

*Chapter organization*: Section 3.3 presents the related work. Section 3.4 presents the main results of this work. Section 3.5 outlines the proof techniques. Section 3.6 presents experimental simulations to demonstrate the practical utility of the established mini-batching limits and tail-averaging. The proofs of all the claims and theorems are provided in the appendix.

## 3.3 Related Work

Stochastic approximation has been the focus of much efforts starting with the work of [99], and has been analyzed in subsequent works including [83, 63] [64, chap. 5,10,11]. These questions and the related issues of computation versus statistics tradeoffs have received renewed attention owing to their relevance in the context of modern large scale machine learning, as highlighted by the work of [12].

*Geometric Rates on initial error:* For *offline optimization* with strongly convex objectives, gradient descent [18] and fast gradient methods [92, 85] indicate linear convergence. However, a multiplicative coupling of number of samples $n$ and condition number in the

[1][25] guarantee these bounds with learning rate $\gamma \rightarrow 0$. This work supports these bounds with $\gamma = 1/R^2$.

[2]Initial error oracle provides initial excess risk $\Delta_0 = L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and noise level $\sigma^2$.

| **Algorithm** | **Final error** | **Runtime/Work** | **Depth** | **Streaming** | **Agnostic** |
|---|---|---|---|---|---|
| Gradient Descent [18] | $\mathcal{O}\left(\frac{\sigma^2 d}{n}\right)$ | $\kappa nd \log \frac{n\cdot\Delta_0}{\sigma^2 d}$ | $\kappa \log \frac{n\cdot\Delta_0}{\sigma^2 d}$ | × | ✓ |
| SDCA [108] | $\mathcal{O}\left(\frac{\sigma^2 d}{n}\right)$ | $(n + \frac{R^2}{\lambda_{\min}} d)d \cdot \log \frac{n\cdot\Delta_0}{\sigma^2 d}$ | $(n + \frac{R^2}{\lambda_{\min}} d) \cdot \log \frac{n\cdot\Delta_0}{\sigma^2 d}$ | × | ✓ |
| Averaged SGD [25][1] | $\mathcal{O}\left(\frac{1}{\lambda_{\min}^2 n^2 \gamma^2} \cdot \Delta_0 \ + \frac{\sigma^2 d}{n}\right)$ | $nd$ | $n$ | ✓ | × |
| Streaming SVRG with initial error oracle [2] [35] | $\mathcal{O}\left(\exp\left(-\frac{n\lambda_{\min}(\mathbf{H})}{R^2}\right) \cdot \Delta_0\right) + \frac{\sigma^2 d}{n}$ | $nd$ | $(\frac{R^2}{\lambda_{\min}(\mathbf{H})}) \cdot \log \frac{n\cdot\Delta_0}{\sigma^2 d}$ | ✓ | ✓ |
| Algorithm 2 (this chapter) | $\mathcal{O}\left(\left(\frac{R^2 t}{\lVert\mathbf{H}\rVert_2 n}\right)^{\frac{t}{\kappa\log(\kappa)}} \cdot \Delta_0 + \frac{\sigma^2 d}{n}\right)$ | $nd$ | $\frac{t}{t-\kappa\log(\kappa)} \cdot \kappa\log(\kappa) \cdot \log\left(\frac{n\cdot\Delta_0}{\sigma^2 d} \cdot \frac{R^2 t}{\lVert\mathbf{H}\rVert_2}\right)$ | ✓ | ✓ |
| Algorithm 2 with initial error oracle (this chapter) | $\mathcal{O}\left(\exp\left(-\frac{n\lambda_{\min}(\mathbf{H})}{R^2\cdot\log(\kappa)}\right) \cdot \Delta_0 + \frac{\sigma^2 d}{n}\right)$ | $nd$ | $\kappa\log(\kappa)\log\frac{n\cdot\Delta_0}{\sigma^2 d}$ | ✓ | ✓ |

Table 3.1: Comparison of Algorithm 2 with existing Algorithms including Gradient Descent, SDCA (offline methods) and averaged SGD, streaming SVRG (streaming methods) given $n$ samples for LSR, with $\Delta_0 = L(\mathbf{w}_0) - L(\mathbf{w}^*)$. The error of offline methods are obtained by running these algorithms so that their final error is $\mathcal{O}(\sigma^2 d/n)$ (which is the minimax rate for the realizable case). The table is written assuming the realizable case; for algorithms which support agnostic case, these bounds can be appropriately modified. Refer to Section 3.2.1 for the definitions of all quantities. We do not consider accelerated variants in this table. Note that the accelerated variants have served to improve running times of the offline algorithms, with the exception of [54]. In the bounds for Algorithm 2, we require $t \geq 24\kappa\log(\kappa)$. Finally, note that streaming SVRG does not conform to the first order oracle ([1]).

computational effort is a major drawback in the large scale context. These limitations are addressed through developments in offline stochastic methods [101, 108, 57, 24] and their accelerated variants [107, 34, 72, 23, 2] which offer near linear running time in the number of samples and condition number with $\log(n)$ passes over the dataset stored in memory.

For *stochastic approximation* with strongly convex objectives, SGD offers linear rates on the bias without achieving minimax rates on the variance [5, 81, 13]. In contrast, iterate averaged SGD [103, 94] offers a sub-linear $\mathcal{O}(1/n^2)$ rate on the bias [25, 29] while achieving minimax rates on the variance. Note that all these results consider the well-specified (additive noise) case when stating the generalization error bounds. We are unaware of any results that provide sharp non-asymptotic analysis of SGD and the related step size issues in the general mis-specified case. Streaming SVRG [35] offers a geometric rate on the bias and optimal statistical error rates; we will return to a discussion of Streaming SVRG below. In terms of methods faster than SGD, our own effort [54] provides the first accelerated stochastic approximation method that improves over SGD on every problem instance.

*Parallelization of Machine Learning algorithms:* In *offline optimization*, [15] study parallel co-ordinate descent for sparse optimization. Parallelization via mini-batching has been studied in [21, 117, 107, 118]. These results compare worst case upper bounds on the training error to argue parallelization speedups, thus providing weak upper bounds on mini-batching limits. Parameter mixing/Model averaging [76] guarantees linear parallelization speedups on the variance but do not improve the bias. Approaches that attempt to re-conciliate communication-computation tradeoffs [71] indicate increased mini-batching hurts convergence, and this is likely an artifact of comparing weak upper bounds. Hogwild [90] indicates near-linear parallelization speedups in the harder asynchronous optimization setting, relying on specific input structures like hard sparsity; these bounds are obtained by comparing worst case upper bounds on training error. Refer to oracle models paragraph below for details on these worst case upper bounds.

In the *stochastic approximation* context, [26] study mini-batching in an oracle model that assumes bounded variance of stochastic gradients. These results compare worst case bounds on the generalization error to prescribe mini-batching limits, which renders these limits to be too loose (as mentioned in their paper). Our chapter's mini-batching result offers guidelines on batch sizes for linear parallelization speedups by comparing generalization bounds that hold on a per problem basis as opposed to worst case bounds. Refer to the paragraph on oracle models for more details. Finally, parameter mixing in the stochastic approximation context [100, 127] offers linear parallelization speedups on the variance error while not improving the bias [100]. Finally, [31] guarantees asymptotic optimality of asynchronous optimization with linear parallelization speedups on the variance.

*Oracle models and optimality:* In stochastic approximation, there are at least two lines of thought with regards to oracle models and notions of optimality. One line involves considering the case of bounded noise [64, chap. 5,10,11], [62, chap. 1,3,6], or, bounded variance of the stochastic gradient, which in the least squares setting amounts to assuming bounds on

$$\widehat{\nabla L}(\mathbf{w}) - \nabla L(\mathbf{w}) = (\mathbf{x}\mathbf{x}^\top - \mathbf{H})(\mathbf{w} - \mathbf{w}^*) - \epsilon\mathbf{x}.$$

This implies additional assumptions are required on compactness of the parameter set (which are enforced via projection steps); such assumptions do not hold in practical implementation of stochastic gradient methods and in the setting considered by this paper. Thus, the mini-batching thresholds in [21, 90, 26, 71] present bounds in the above worst-case oracle model by comparing weak upper bounds on the training/test error.

Another view of optimality [4, 33] considers an objective where the goal is to match the rate of the statistically optimal estimator (referred to as the $M-$estimator) on every problem instance. [94] consider this oracle model for the LSR problem and prove that the distribution of the averaged SGD estimator on every problem matches that of the $M-$estimator under certain regularity conditions [69, chap. $5-6$]. A recent line of work [7, 35] aims to provide non-asymptotic guarantees for SGD and its variants in this oracle model. This chapter aims to understand mini-batching and other computational aspects of parallelizing stochastic approximation on every problem instance by working in this practically relevant oracle model. Refer to [54] for more details.

*Comparing offline and streaming algorithms:* Firstly, offline algorithms require performing multiple passes over a dataset stored in memory. Note that results and convergence rates established in the finite sum/offline optimization context do not translate to rates on the generalization error. Indeed, these results require going though concentration and a generalization error analysis for this translation to occur. Refer to [35] for more details.

*Comparison to streaming SVRG:* Streaming SVRG does not function in the stochastic first order oracle model [1] satisfied by SGD as run in practice since it requires gradients at two points from a single sample [35]. Furthermore, in contrast to this work, its depth bounds depend on a stronger fourth moment property due to lack of mini-batching.

## 3.4 Main Results

We begin by writing out the behavior of the learning rate as a function of batch size.

*Maximal Learning Rates:* We write out a characterization of the largest learning rate $\gamma_{b,\max}^{div}$ that permits the convergence of the mini-batch Stochastic Gradient Descent update. The following generalized eigenvector problem allows for the computation of $\gamma_{b,\max}^{div}$:

$$\frac{2}{\gamma_{b,\max}^{div}} = \sup_{\mathbf{W}\in\mathcal{S}(d)} \frac{\langle \mathbf{W}, \mathcal{M}\mathbf{W}\rangle + (b-1)\cdot \operatorname{Tr}\mathbf{WHWH}}{b\cdot \operatorname{Tr}\mathbf{WHW}}. \tag{3.2}$$

This characterization generalizes the divergent stepsize characterization of [25] for batch sizes $> 1$. The derivation of the above characterization can be found in appendix A.5.1. We note that this characterization sheds light on how the divergent learning rates interpolate from batch size 1 (which is $\leq 2/\operatorname{Tr}\mathbf{H}$) to the batch gradient descent learning rate (setting $b$ to $\infty$), which turns out to be $2/\lambda_{\max}(\mathbf{H})$. A property of $\gamma_{b,\max}^{div}$ worth noting is that it does not depend on properties of the noise ($\mathbf{\Sigma}$), and depends only on the second and fourth moment properties of the covariate $\mathbf{x}$.

We note that in this chapter, our interest does not lie in the non-divergent stepsizes $0 \leq \gamma \leq \gamma_{b,\max}^{div}$, but in the set of (maximal) stepsizes $0 \leq \gamma \leq \gamma_{b,\max}$ ($< \gamma_{b,\max}^{div}$) that are sufficient to guarantee minimax error rates of $\mathcal{O}(\widehat{\sigma^2_{\text{MLE}}/n})$. For the LSR problem, these maximal learning rates $\gamma_{b,\max}$ are:

$$\gamma_{b,\max} \overset{\text{def}}{=} \frac{2b}{R^2\cdot\rho_{\text{m}} + (b-1)\|\mathbf{H}\|_2}, \text{ where, } \rho_{\text{m}} \overset{\text{def}}{=} \frac{d\|(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\|_2}{\operatorname{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right)}. \tag{3.3}$$

Note that $\rho_{\text{m}} \geq 1$ captures a notion of "degree" of model mismatch, and how it impacts the learning rate $\gamma_{b,\max}$; for the additive noise/well specified/homoscedastic case, $\rho_{\text{m}} = 1$. Thus, for problems where $R^2$ and $\|\mathbf{H}\|_2$ is held the same, the well-specified variant of the LSR problem admits a strictly larger learning rate (that achieves minimax rates on the variance) compared to the mis-specified case. Furthermore, in stark contrast to the well-specified case, $\gamma_{b,\max}$ in the mis-specified case depends not just on the second and fourth moment properties of the input, but also on the noise covariance $\mathbf{\Sigma}$. We show that our characterization of $\gamma_{b,\max}$ in the mis-specified case is tight in that there exist problem instances where $\gamma_{b,\max}$ (equation (3.3)) is off the maximal learning rate in the well-specified case (obtained by setting $\rho_{\mathrm{m}} = 1$ in equation (3.3)) by a factor of the dimension $d$ and $\gamma_{b,\max}$ is still the largest step size yielding minimax rates. We also note that there could exist mis-specified problem instances where a step size $\gamma$ exceeding $\gamma_{b,\max}$ achieves minimax rates. Characterizing the maximal learning rate that achieves minimax rates on *every mis-specified* problem instance is an interesting open question. We return to the characterization of $\gamma_{b,\max}$ in section 3.4.1.

Note that this chapter characterizes the performance of Algorithms 1 and 2 when run with a step size $\gamma \leq \frac{\gamma_{b,\max}}{2}$. The proofs turn out to be tedious for $\gamma \in \left(\frac{\gamma_{b,\max}}{2}, \gamma_{b,\max}\right)$ and can be found in the initial version of this paper [55] and these were obtained through generalizing the operator view of analyzing SGD methods introduced by [25]. For the well-specified case, this chapter's results hold for the same learning rate regimes as [7, 35], that are known to admit statistical optimality. We also note that in the additive noise case, we are unaware of a separation between $\gamma_{b,\max}$ and $\gamma_{b,\max}^{div}$; but as we will see, this is not of much consequence given that there exists a strict separation in the learning rate $\gamma_{b,\max}$ between the well-specified and mis-specified problem instances.

Finally, note that the stochastic process viewpoint allows us to work with learning rates that are significantly larger compared to standard analyses that use function value contraction e.g., [13, Theorem 4.6]. All existing works establishing mini-batching thresholds in the stochastic optimization setting e.g., [26] work in the worst case (bounded noise) oracle with small step sizes, and draw conclusions on mini-batch thresholds and effects by comparing weak upper bounds on the excess risk.

*Mini-Batched Tail-Averaged SGD for the mis-specified case:* We present our main result, which is the error bound for mini-batch tail-averaged SGD for the general mis-specified LSR problem.

**Algorithm 1** Minibatch-TailAveraging-SGD

```
Input: Initial point $\mathbf{w}_0$, stepsize $\gamma$, minibatch size $b$, initial iterations $s$, total samples $n$.
1: for $t = 1, 2, .., \lfloor \frac{n}{b} \rfloor$ do
2:     Sample "$b$" tuples $\{(x_{ti}, y_{ti})\}_{i=1}^{b} \sim \mathcal{D}^b$
3:     $\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \frac{\gamma}{b} \sum_{i=1}^{b} \widehat{\nabla L_{ti}}(\mathbf{w}_{t-1})$
4: end for
Output: $\bar{\mathbf{w}} = \frac{1}{\lfloor \frac{n}{b} \rfloor - s} \sum_{i>s} \mathbf{w}_i$
```

**Theorem 1.** *Consider the general mis-specified case of the LSR problem* (3.1)*. Running Algorithm 1 with a batch size $b \geq 1$, step size $\gamma \leq \gamma_{b,max}/2$, number of unaveraged iterations $s$, total number of samples $n$, we obtain an iterate $\overline{\mathbf{w}}$ satisfying the following excess risk bound:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \frac{2}{\gamma^2\mu^2} \cdot \frac{(1-\gamma\mu)^s}{\left(\frac{n}{b} - s\right)^2} \cdot \big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\big) + 4 \cdot \frac{\widehat{\sigma^2_{MLE}}}{b \cdot (\frac{n}{b} - s)}. \tag{3.4}$$

*In particular, with $\gamma = \gamma_{b,max}/2$, we have the following excess risk bound:*

$$L(\overline{\mathbf{w}}) - L(\mathbf{w}^*) \leq \underbrace{\frac{2\kappa_b^2}{\left(\frac{n}{b} - s\right)^2} \exp\left(-\frac{s}{\kappa_b}\right) \big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\big)}_{\mathfrak{T}_1} + \underbrace{4 \cdot \frac{\widehat{\sigma^2_{MLE}}}{b(\frac{n}{b} - s)}}_{\mathfrak{T}_2},$$

*with $\kappa_b = \frac{R^2 \cdot \rho_m + (b-1)\|\mathbf{H}\|_2}{b\lambda_{min}(\mathbf{H})}$.*

Note that the above theorem indicates that the excess risk is composed of two terms, namely the bias ($\mathfrak{T}_1$), which represents the dependence on the initial conditions $\mathbf{w}_0$ and the variance ($\mathfrak{T}_2$), which depends on the statistical noise ($\widehat{\sigma^2_{\mathrm{MLE}}}$); the bias decays geometrically during the "$s$" unaveraged iterations while the variance is minimax optimal (up to constants) provided $s = \mathcal{O}(n)$. We will understand this geometric decay on the bias more precisely.

*Effect of tail-averaging SGD's iterates:* To understand tail-averaging, we specialize theorem 1 with a batch size 1 to the well-specified case, i.e., where, $\mathbf{\Sigma} = \sigma^2\mathbf{H}$, $\widehat{\sigma^2_{\mathrm{MLE}}} = d\sigma^2$ and $\rho_{\mathrm{m}} = 1$.

**Corollary 2.** *Consider the well-specified (additive noise) case of the streaming LSR problem ($\mathbf{\Sigma} = \sigma^2 \mathbf{H}$), with a batch size $b = 1$. With a learning rate $\gamma = \frac{\gamma_{1,max}}{2} = \frac{1}{R^2}$, unaveraged iterations $s$ and total samples $n$, we have the following excess risk bound:*

$$L(\overline{\mathbf{w}}) - L(\mathbf{w}^*) \leq \underbrace{\frac{2\kappa_1^2}{(n-s)^2} \exp\left(-\frac{s}{\kappa_1}\right) \{L(\mathbf{w}_0) - L(\mathbf{w}^*)\}}_{\mathfrak{T}_1} + \underbrace{4 \cdot \frac{d\sigma^2}{n-s}}_{\mathfrak{T}_2}, \textit{ where, } \kappa_1 = R^2/\mu.$$

Tail-averaging allows for a geometric decay of the initial error $\mathfrak{T}_1$, while tail-averaging over $s = c{\cdot}n$ (with $c < 1$), allows for the variance $\mathfrak{T}_2$ to be minimax optimal (up to constants). We note that the work of [79], which studies empirical optimization for training non-convex sequence models (e.g. Long-Short term memory models (LSTMs)) also indicate the benefits of tail-averaging.

Note that this particular case (i.e. additive noise/well-specified case with batch size 1) with tail-averaging from start ($s = 0$) is precisely the setting considered in [25], and their result (a) achieves a sub-linear $\mathcal{O}(1/n^2)$ rate on the bias and (b) their variance term is shown to be minimax optimal only with learning rates that approach zero (i.e. $\gamma \to 0$).

### *3.4.1 Effects Of Learning Rate, Batch Size and The Role of Mis-specified Models*

We now consider the interplay of learning rate, batch size and how model mis-specification plays into the mix. Towards this, we split this section into three parts: (a) understanding learning rate versus mini-batch size in the well-specified case, (b) how model mis-specification leads to a significant difference in the behavior of SGD and (c) how model mis-specification manifests itself when considered in tradeoff between the learning rate versus batch-size.

*Effects of mini-batching in the well-specified case:* As mentioned previously, in the well-specified case, $\mathbf{\Sigma} = \sigma^2 \mathbf{H}$ and $\rho_{\mathrm{m}} = 1$. For this case, equation (3.3) can be specialized as:

$$\gamma_{b,\max} = \frac{2b}{R^2 + (b-1)\|\mathbf{H}\|_2}. \tag{3.5}$$

Observe that the learning rate $\gamma_{b,\max}$ grows linearly as a function of the batch size $b$ until a batch size $b = b_{\mathrm{thresh}} = 1 + \frac{R^2}{\|\mathbf{H}\|_2}$. In the regime of batch sizes $1 < b \leq b_{\mathrm{thresh}}$, the resulting mini-batch SGD updates offer near-linear parallelization speedups over SGD with a batch size of 1. Furthermore, increasing batch sizes beyond $b_{\text{thresh}}$ leads to sub-linear increase in the learning rate, and this implies that we lose the linear parallelization speedup offered by mini-batching with a batch-size $b \leq b_{\text{thresh}}$. Losing the linear parallelization is indicative of the following: consider the case when we double batch-size from $b > b_{\text{thresh}}$ to $2b$. Suppose the bias error $\mathfrak{T}_1$ is larger than the variance $\mathfrak{T}_2$, we require performing the same number of updates with a batch size $2b$ as we did with a batch size $b$ to achieve a similar excess risk bound; this implies we are inefficient in terms of number of samples (or, number of gradient computations) used to achieve a given excess risk. When the estimation error ($\mathfrak{T}_2$) dominates the approximation error ($\mathfrak{T}_1$), we note that larger batch sizes $b$ (with $b > b_{\text{thresh}}$) serves to improve the variance term, thus allowing linear parallelization speedups via mini-batching.

Note that with a batch size of $b = b_{\text{thresh}}$, the learning rate of $\mathcal{O}(1/\lambda_{\max}(\mathbf{H}))$ employed by mini-batch SGD resembles ones used by batch gradient descent. This mini-batching characterization thus allows for understanding tradeoffs of learning rate versus batch size. This behavior is noted in practice (empirically, but with no underlying rigorous theory) for a variety of problems (going beyond linear regression/convex optimization), in the deep learning context [40].

*SGD's behaviour with mis-specified models:* Next, this chapter attempts to shed light on some fundamental differences in the behavior of SGD when dealing with the mis-specified case (as against the well-specified case, which is the focus of existing results [94, 7, 29, 25]) of the LSR problem. This chapter's results in general mis-specified case with batch sizes $b > 1$ specialize to existing results additive noise/well-specified case with batch size 1 [7, 29]. To understand these issues better, we consider $\gamma_{b,\max}$ in equation (3.3) with a batch size 1:

$$\gamma_{1,\max} = \frac{2}{R^2 \cdot \rho_{\mathrm{m}}}. \tag{3.6}$$

Recounting that $\rho_{\mathrm{m}} \geq 1$, observe that the mis-specified case admits a maximal learning rate (with a view of achieving minimax rates) that is at most as large as the additive noise/well-specified case, where $\rho_{\mathrm{m}} = 1$. Note that when $\mathrm{Tr}\,(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma})$ is nearly the same (say, upto constants) as the spectral norm $\|(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\|_2$, then $\rho_{\mathrm{m}} = \mathcal{O}(d)$ and $\gamma_{1,\max} = \mathcal{O}(\frac{1}{R^2 d})$. This implies that there exist mis-specified models whose noise properties (captured through the noise covariance matrix $\mathbf{\Sigma}$) prevents SGD from working with large learning rates of $\mathcal{O}(1/R^2)$ used in the well-specified case.

This notion is formalized in the following lemma, which presents an instance working with the mis-specified case, wherein, SGD *cannot* employ large learning rates used by the well-specified variant of the problem, while *retaining minimax optimality.* This behavior is in stark contrast to algorithms such as streaming SVRG ([35]), which work with the same large learning rates in the mis-specified case as in the well-specified case, while guaranteeing minimax optimal rates. The proof of Lemma 3 can be found in the appendix A.5.6.

**Lemma 3.** *Consider a Streaming LSR example with Gaussian covariates (i.e.* $\mathbf{x} \sim \mathcal{N}(0, \mathbf{H})$*) with a diagonal second moment matrix* $\mathbf{H}$ *that is defined by:*

$$\mathbf{H}_{ii} = \begin{cases} 1 & \text{if } i = 1 \\ 1/d & \text{if } i > 1 \end{cases}.$$

*Further, let the noise covariance matrix* $\mathbf{\Sigma}$ *be diagonal as well, with the following entries:*

$$\mathbf{\Sigma}_{ii} = \begin{cases} 1 & \text{if } i = 1 \\ 1/[(d-1)d] & \text{if } i > 1 \end{cases}.$$

*For this problem instance,* $\gamma_{1,max} \le \frac{4}{(d+2)(1+\frac{1}{d})}$ *is necessary for retaining minimax rates, while the well-specified variant of this problem permits a maximal learning rate* $\le \frac{d}{(d+2)(1+\frac{1}{d})}$*, thus implying an* $\mathcal{O}(d)$ *separation in learning rates between the well-specified and mis-specified case.*

*Learning rate versus mini-batch size issues in the mis-specified case:* Noting that for the batch size 1, as mentioned in equation (3.6), the learning rate for the mis-specified case in the most optimistic situation (when $\rho_{\mathrm{m}} =$ constant) can be atmost as large as the learning rate for the well-specified case. Furthermore, we also know from the observations in the mis-specified case that the learning rate tends to grow linearly as a function of the batch size until it hits the limit of $\mathcal{O}(1/\lambda_{\max}(\mathbf{H}))$. Combining these observations, we will revisit equation (3.3), which says:

$$\gamma_{b,\max} \overset{\text{def}}{=} \frac{2b}{R^2 \cdot \rho_{\text{m}} + (b-1)\|\mathbf{H}\|_2}.$$

This implies that the mini-batching size threshold $b_{\text{thresh}}$ can be expressed as:

$$b_{\text{thresh}} \overset{\text{def}}{=} 1 + \frac{R^2}{\|\mathbf{H}\|_2} \cdot \rho_{\text{m}}. \tag{3.7}$$

When $1 < b \leq b_{\text{thresh}}$, we achieve near linear parallelization speedups over running SGD with a batch size 1. Note that this characterization specializes to the batch size threshold $b_{\text{thresh}}$ presented in the well-specified case (i.e. where $\rho_{\text{m}} = 1$). Furthermore, this batch size threshold (in the mis-specified case) could be much larger than the threshold in the well-specified case, which is expected since the learning rate for a batch size 1 in the mis-specified case can potentially be much smaller than ones used in the well specified case. Furthermore, with a batch size $b_{\text{thresh}}$, note that the learning rate is $\mathcal{O}(1/\lambda_{\max}(\mathbf{H}))$, resembling ones used with batch gradient descent.

*Behavior of the final-iterate:* We now present the excess risk bound offered by the final iterate of a stochastic gradient scheme. This result is of much practical relevance in the context of modern machine learning and deep learning, where final iterate is often used, and where the tradeoffs between learning rate and batch sizes are discussed in great detail [114]. For this discussion, we consider the well-specified case to present our results owing to its ease in presentation. Our framework and results are generic for translating these observations to the mis-specified case.

**Lemma 4.** *Consider the well-specified case of the LSR problem. Running Algorithm 1 with a step size $\gamma \leq \frac{\gamma_{b,max}}{2} = \frac{b}{R^2+(b-1)\|\mathbf{H}\|_2}$, batch size $b$, total samples $n$ and with no iterate averaging (i.e. with $s = n-1$) yields a result $\mathbf{w}_{\lfloor n/b \rfloor}$ that satisfies the following excess risk bound:*

$$\mathbb{E}\left[L(\mathbf{w}_{\lfloor n/b \rfloor})\right] - L(\mathbf{w}^*) \leq \kappa_b (1-\gamma\mu)^{\lfloor n/b \rfloor} \Big( L(\mathbf{w}_0) - L(\mathbf{w}^*) \Big) + \frac{\gamma}{b}\sigma^2 \operatorname{Tr}(\mathbf{H}), \tag{3.8}$$

*where* $\kappa_b \overset{\text{def}}{=} \frac{R^2+(b-1)\|\mathbf{H}\|_2}{b\mu}$*. In particular, with a step size* $\gamma = \frac{\gamma_{b,max}}{2} = \frac{b}{R^2+(b-1)\|\mathbf{H}\|_2}$*, we have:*

$$\mathbb{E}\left[L(\mathbf{w}_{\lfloor n/b \rfloor})\right] - L(\mathbf{w}^*) \leq \kappa_b \cdot e^{-\frac{\lfloor n/b \rfloor}{\kappa_b}} \cdot \Big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\Big) + \frac{\sigma^2 \operatorname{Tr}(\mathbf{H})}{R^2 + (b-1)\|\mathbf{H}\|_2}. \tag{3.9}$$

*Remarks:* Noting that $\operatorname{Tr}(\mathbf{H}) \leq R^2$, the variance of the final iterate with batch size 1 is $\leq \sigma^2$. Next, with a batch size $b = b_{\text{thresh}}$, the final iterate has a variance $\leq \sigma^2/2$; at cursory glance this may appear interesting, in that by mini-batching, we do not appear to gain much in terms of the variance. This is unsurprising given that in the regime of $b \leq b_{\text{thresh}}$, the $\gamma_{b,\max}$ grows linearly, thus nullifying the effect of averaging multiple stochastic gradients. Furthermore, this follows in accordance with the linear parallelization speedups offered by a batch size $1 < b \leq b_{\text{thresh}}$. Note however, once $b > b_{\text{thresh}}$, any subsequent increase in batch sizes allows the variance of the final iterate to behave as $\mathcal{O}(\sigma^2/b)$. Finally, note that once $b > b_{\text{thresh}}$, doubling batch sizes $b$ (in equation (3.9)) possesses the same effect as halving learning rate from $\gamma$ to $\gamma/2$ (as seen from equation (3.8)), providing theoretical rigor to issues explored in training practical deep models [114].

### *3.4.2 Parallelization via Doubling Batch Sizes and Model Averaging*

We now elaborate on a highly parallelizable stochastic gradient method, which is epoch based and relies on doubling batch sizes across epochs to yield an algorithm that offers the same generalization error as that of offline (batch) gradient descent in nearly the $s$ame number of serial updates as batch gradient descent, while being a streaming algorithm that does not require storing the entire dataset in memory. Following this, we present a non-asymptotic bound for parameter mixing/model averaging, which is a communication efficient parallelization scheme that has favorable properties when the estimation error (i.e. variance) is the dominating term of the excess risk.

*(Nearly) Matching the depth of Batch Gradient Descent:* The result of Theorem 1 establishes a scalar generalization error bound of Algorithm 1 for the general mis-specified case of LSR and showed that the depth (number of sequential updates in our algorithm) is decreased to $n/b$. This section builds upon this result to present a simple and intuitive doubling

**Algorithm 2** MinibatchDoublingPartialAveragingSGD

```
Input: Initial point $\mathbf{w}_0$, stepsize $\gamma$, initial minibatch size $b$, number of iterations in each epoch $s$, number of samples $n$.
1: /*Run logarithmic number of epochs where each epoch runs $t$ iterations of minibatch SGD (with out averaging). Double minibatch size after each epoch.*/
2: for $\ell = 1, 2, \cdots, \log \frac{n}{bt} - 1$ do
3:     $b_\ell \leftarrow 2^{\ell-1} b$
4:     $\mathbf{w}_\ell \leftarrow$ Minibatch-TailAveraging-SGD$(\mathbf{w}_{\ell-1}, \gamma, b_\ell, t-1, t \cdot b_\ell)$
5: end for
6: /*For the last epoch, run tail averaged minibatch SGD with initial point $\mathbf{w}_t$, stepsize $\gamma$, minibatch size $2^{\log \frac{n}{bt} - 1} \cdot b = n/2t$, number of initial iterations $t/2$ and number of samples $n/2$.*/
7: $\overline{\mathbf{w}} \leftarrow$ Minibatch-TailAveraging-SGD$(\mathbf{w}_s, \gamma, n/2t, t/2, n/2)$
Output: $\overline{\mathbf{w}}$
```

based streaming algorithm that works in epochs and processes a total of $n/2$ points. In each epoch, the minibatch size is increased by a factor of 2 while applying Algorithm 1 (with no tail-averaging) with twice as many samples as the previous epoch. After running over $n/2$ samples using this epoch based approach, we run Algorithm 1 (with tail-averaging) with the remaining $n/2$ points. Note that each epoch decays the bias of the previous epoch linearly and halves the statistical error (since we double mini-batch size). The final tail-averaging phase ensures that the variance is small.

The next theorem formalizes this intuition and shows Algorithm 2 improves the depth exponentially from $n/b_{\text{thresh}}$ to $\mathcal{O}\left(\kappa \log(d\kappa) \log(n\{L(\mathbf{w}_0) - L(\mathbf{w}^*)\}/\widehat{\sigma^2_{\text{MLE}}})\right)$ in the presence of an error oracle that provides us with the initial excess risk $L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and the noise level $\widehat{\sigma^2_{\text{MLE}}}$.

**Theorem 5.** *Consider the general mis-specified case of LSR. Suppose in Algorithm 2, we* *use initial batchsize of $b = b_{thresh}$, stepsize $\gamma = \frac{\gamma_{b,max}}{2}$ and number of iterations in each epoch being $t \geq 24\kappa \log(\kappa)$, we obtain the following excess risk bound on $\overline{\mathbf{w}}$:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \left(\frac{2bt}{n}\right)^{\frac{t}{12\kappa\log(\kappa)}} \cdot \big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\big) + 80\frac{\widehat{\sigma^2_{MLE}}}{n}.$$

**Remarks**: The final error again has two parts: the bias term that depends on the initial error $L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and the variance term that depends on the statistical noise $\widehat{\sigma^2_{\text{MLE}}}$. Note that the variance error decays at a rate of $\mathcal{O}\left(\widehat{\sigma^2_{\text{MLE}}}/n\right)$ which is minimax optimal up to constant factors.

Algorithm 2 decays the bias at a superpolynomial rate by choosing $t$ large enough. If Algorithm 2 has access to an initial error oracle that provides $L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and $\widehat{\sigma^2_{\text{MLE}}}$, we can run Algorithm 2 with a batch size $b_{\text{thresh}}$ until the excess risk drops to the noise level $\widehat{\sigma^2_{\text{MLE}}}$ and subsequently begin doubling the batch size. Such an algorithm indeed gives geometric convergence with a generalization error bound as:

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \exp\left(-(\frac{n\lambda_{\min}}{R^2 \cdot \log(\kappa)}) \cdot \frac{1}{\rho_{\text{m}}}\right)\{L(\mathbf{w}_0) - L(\mathbf{w}^*)\} + 80\frac{\widehat{\sigma^2_{\text{MLE}}}}{n},$$

with a depth of $\mathcal{O}\left(\kappa \log(d\kappa) \log \frac{n\{L(\mathbf{w}_0)-L(\mathbf{w}^*)\}}{\widehat{\sigma^2_{\text{MLE}}}}\right)$. The proof of this claim follows relatively straightforwardly from the proof of Theorem 5. We note that this depth nearly matches (up to log factors), the depth of standard offline gradient descent despite being a streaming algorithm. This algorithm (aside from tail-averaging in the final epoch) resembles empirically effective schemes proposed in the context of training deep models [114].

*Parameter Mixing/Model-Averaging:* We consider a communication efficient method for distributed optimization which involves running mini-batch tail-averaged SGD independently on $P$ separate machines (each containing their own independent samples) and averaging the resulting solution estimates. This is a well studied scheme for distributed optimization [76, 130, 100, 127]. As mentioned in [100], these schemes do not appear to offer improvements in the bias error while offering near linear parallelization speedups on the variance. We provide here a non-asymptotic characterization of the behavior of model averaging for the general mis-specified LSR problem.

**Theorem 6.** *Consider running Algorithm 1, i.e., mini-batch tail-averaged SGD (for the mis-specified LSR problem* (3.1)*) independently in $P$ machines, each of which contains $N/P$ samples. Let Algorithm 1 be run with a batch size $b$, learning rate $\gamma \leq \gamma_{b,max}/2$, tail-averaging begun after $s-$iterations, and let each of these machines output $\{\overline{\mathbf{w}}_i\}_{i=1}^{P}$. The excess risk of the model-averaged estimator $\overline{\mathbf{w}} = \frac{1}{P}\sum_{i=1}^{P}\overline{\mathbf{w}}_i$ is upper bounded as:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \frac{(1-\gamma\mu)^s}{\gamma^2\mu^2\left(\frac{n}{P\cdot b} - s\right)^2} \cdot \frac{2 + (P-1)(1-\gamma\mu)^s}{P} \cdot \Big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\Big) \\ + 4 \cdot \frac{\widehat{\sigma^2_{MLE}}}{P \cdot b \cdot \left(\frac{n}{P\cdot b} - s\right)}.$$

*In particular, with $\gamma = \gamma_{b,max}/2$, we have the following excess risk bound:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \exp\left(-\frac{s}{\kappa_b}\right) \cdot \frac{\kappa_b^2}{\left(\frac{n}{P\cdot b} - s\right)^2} \cdot \frac{2 + (P-1)\cdot\exp(-s/\kappa_b)}{P} \cdot \big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\big) \\ + 4 \cdot \frac{\widehat{\sigma^2_{MLE}}}{P \cdot b \cdot \left(\frac{n}{P\cdot b} - s\right)}.$$

*Remarks:* We note that during the iterate-averaged phase (i.e. $t > s$), there is no reduction of the bias, whereas, during the (initial) unaveraged iterations, once $s > \kappa_b \log(P)$, we achieve linear speedups on the bias. We note that model averaging offers linear parallelization speedups on the variance error. Furthermore, when the bias reduces to the noise level, model averaging offers linear parallelization speedups on the overall excess risk. Note that if $s = c \cdot n/(P \cdot b)$, with $c < 1$, then the excess risk is minimax optimal. Finally, we note that the theorem can be generalized in a straightforward manner to the situation when each machine has different number of examples.

## 3.5 Proof Outline

We present here the framework for obtaining the results described in this chapter; the framework has been introduced in the work of [25]. Towards this purpose, we begin by introducing some notations. We begin by defining the centered estimate $\boldsymbol{\eta}_t$ as:

$$\boldsymbol{\eta}_t \overset{\text{def}}{=} \mathbf{w}_t - \mathbf{w}^*.$$

Mini-batch SGD (with a batch size $b$) moves $\boldsymbol{\eta}_{t-1}$ to $\boldsymbol{\eta}_t$ using the following update:

$$\boldsymbol{\eta}_t = \left(\mathbf{I} - \frac{\gamma}{b} \cdot \sum_{i=1}^{b} \mathbf{x}_{ti} \otimes \mathbf{x}_{ti}\right)\boldsymbol{\eta}_{t-1} + \frac{\gamma}{b}\sum_{i=1}^{b} \epsilon_{ti}\mathbf{x}_{ti} = (\mathbf{I} - \gamma\widehat{\mathbf{H}}_{tb})\boldsymbol{\eta}_{t-1} + \gamma \cdot \boldsymbol{\xi}_{tb},$$

where, $\widehat{\mathbf{H}}_{tb} = \frac{1}{b}\sum_{i=1}^{b} \mathbf{x}_{ti} \otimes \mathbf{x}_{ti}$ and $\boldsymbol{\xi}_{tb} = \frac{1}{b}\sum_{i=1}^{b} \epsilon_{ti}\mathbf{x}_{ti}$. Next, the tail-averaged iterate $\bar{\mathbf{x}}_{s,n}$ is associated with its own centered estimate $\bar{\boldsymbol{\eta}}_{s,n} = \frac{1}{n-s}\sum_{i=s+1}^{n} \boldsymbol{\eta}_i$. The analysis proceeds by tracking the covariance of the centered estimates $\boldsymbol{\eta}_t$, i.e. by tracking $\mathbb{E}\left[\boldsymbol{\eta}_t \otimes \boldsymbol{\eta}_t\right]$.

*Bias-Variance decomposition:* The main results of this chapter are derived by going through the bias-variance decomposition, which is well known in the context of Stochastic Approximation [5, 7, 35]. The bias-variance decomposition allows for us to bound the generalization error by analyzing two sub-problems, namely, (i) The *bias* sub-problem, which analyzes the noiseless/realizable (or the consistent linear system) problem, by setting the noise $\epsilon_{ti} = 0 \ \forall \ t, i$, $\boldsymbol{\eta}_0^{\text{bias}} = \boldsymbol{\eta}_0$ and (ii) the *variance* sub-problem, which involves starting at the solution, i.e., $\boldsymbol{\eta}_0^{\text{variance}} = 0$ and allowing the noise $\epsilon_{ti}$ to drive the resulting process. The corresponding tail-averaged iterates are associated with their centered estimates $\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}$ and $\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}$ respectively. The bias-variance decomposition for the square loss establishes the following relation:

$$\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n} \otimes \bar{\boldsymbol{\eta}}_{s,n}\right] \preceq 2 \cdot \left(\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}} \otimes \bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}\right] + \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}} \otimes \bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}\right]\right). \tag{3.10}$$

Using the bias-variance decomposition, we obtain an estimate of the generalization error as

$$\begin{aligned}\mathbb{E}\left[L(\bar{\mathbf{x}}_{s,n})\right] - L(\mathbf{x}^*) &= \tfrac{1}{2} \cdot \langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n} \otimes \bar{\boldsymbol{\eta}}_{s,n}\right]\rangle \\ &\leq \operatorname{Tr}\left(\mathbf{H} \cdot \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}} \otimes \bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}\right]\right) + \operatorname{Tr}\left(\mathbf{H} \cdot \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}} \otimes \bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}\right]\right).\end{aligned}$$

We now provide a few lemmas that help us bound the behavior of the bias and variance error.

**Lemma 7.** *With a batch size $b$, step size $\gamma = \gamma_{b,max}/2$, the centered bias estimate $\boldsymbol{\eta}_t^{bias}$ exhibits the following per step contraction:*

$$\langle \mathbf{I}, \mathbb{E}\left[\boldsymbol{\eta}_t^{bias} \otimes \boldsymbol{\eta}_t^{bias}\right]\rangle \leq c_{\kappa_b}\langle \mathbf{I}, \mathbb{E}\left[\boldsymbol{\eta}_{t-1}^{bias} \otimes \boldsymbol{\eta}_{t-1}^{bias}\right]\rangle,$$

*where,* $c_{\kappa_b} = 1 - 1/\kappa_b$*, where* $\kappa_b = \frac{R^2 \cdot \rho_m + (b-1)\|\mathbf{H}\|_2}{b\mu}$.

Lemma 7 ensures that the bias decays at a geometric rate during the burn-in iterations when the iterates are not averaged; this rate holds only when the excess risk is larger than the noise level $\sigma^2$.

We now turn to bounding the variance error. It turns out that it suffices to understand the behavior of limiting centered variance $\mathbb{E}\left[\boldsymbol{\eta}_\infty^{\text{variance}} \otimes \boldsymbol{\eta}_\infty^{\text{variance}}\right]$.

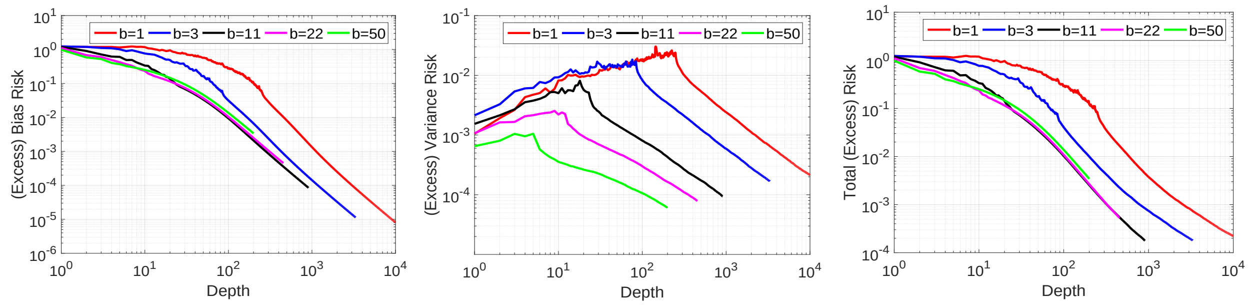
**Lemma 8.** *Consider the well-specified case of the streaming LSR problem. With a batch size* $b$*, step size* $\gamma = \gamma_{b,max}/2$*, the limiting centered variance* $\boldsymbol{\eta}_\infty^{variance}$ *has an expected covariance that is upper bounded in a psd sense as:*

$$\mathbb{E}\left[\boldsymbol{\eta}_\infty^{variance} \otimes \boldsymbol{\eta}_\infty^{variance}\right] \preceq \frac{1}{R^2 + (b-1)\|\mathbf{H}\|_2} \cdot \sigma^2 \cdot \mathbf{I}.$$

Characterizing the behavior of the final iterate is crucial towards obtaining bounds on the behavior of the tail-averaged iterate. In particular, the final iterate having a excess variance risk $\mathcal{O}(\sigma^2)$ (as is the case with Lemma 8) appears crucial towards achieving minimax rates of the averaged iterate.

## 3.6 Experimental Simulations

We conduct experiments using a synthetic example to illustrate the implications of our theoretical results on mini-batching and tail-averaging. The data is sampled from a 50−dimensional Gaussian with eigenvalues decaying as $\{\frac{1}{k}\}_{k=1}^{50}$ (condition number $\kappa = 50$), and the variance $\sigma^2$ of the (additive noise) noise is 0.01. In this case, our estimated batch size according to Theorem 1 is $b_{\text{thresh}} = 11$. Our results are presented by averaging over 100 independent runs of the Algorithm, and each run employs $200\kappa$ samples. All plots are log-log with x-axis being the depth, and y-axis the excess risk. For our plots, we assume that each iteration takes constant time for all batch sizes; this is done to present evidence regarding the tightness of our mini-batching characterization limits that yield linear parallelization speedups over standard serial SGD.

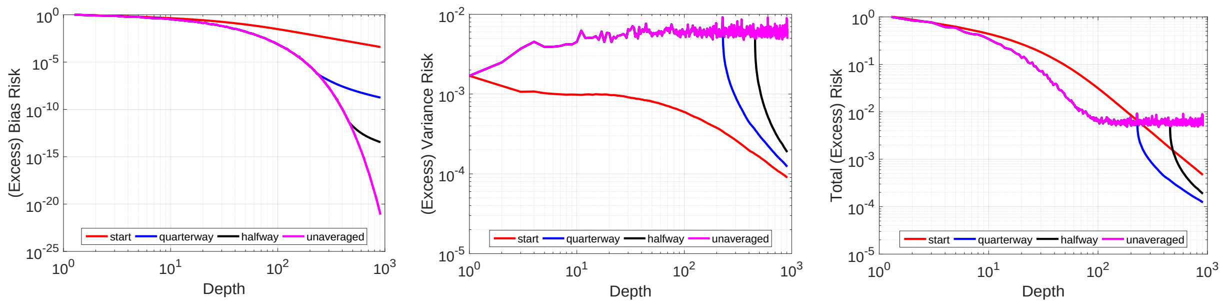


(a) Bias Risk (b) Variance Risk (c) Total Risk

Figure 3.1: Effect of increased batch sizes on the Algorithm's generalization error. The variance decreases monotonically with increasing batch size. The bias indicates that the rate of decay increases till the optimal $b_{thresh}$. With $b = b_{\text{thresh}}$, mini-batch SGD obtains the same generalization error as batchsize 1 using smaller number of iterations (i.e. smaller depth) compared to larger batch sizes.

We consider the effect of mini-batching (in figure 3.1) with batch sizes of 1, 3, $b_{\text{thresh}} = 11$, $2 \cdot b_{\text{thresh}} = 22$ and $d = 50$. Averaging begins after observing a fixed number of samples (set as $5\kappa$). We see that the rate of bias decay (figure 3.1a) increases until reaching a mini-batch size of $b_{\text{thresh}}$, saturating thereafter; this implies we are inefficient in terms of sample size. As expected, the rate of decay of variance (figure 3.1b) is monotonic as a function of mini-batch size. Finally, the overall error (figure 3.1c) shows the tightness of our mini-batching characterization: with a batch size of $b_{\text{thresh}}$, we obtain a generalization error that is the *same* as using batch size of 1 with the number of (serial) iterations (i.e. depth) that is an order of magnitude smaller. Subsequently, we note that larger batch sizes worsen generalization error thus depicting the tightness of our characterization of $b_{\text{thresh}}$.

In the next experiment, we fix batch size $= b_{\text{thresh}}$ and consider the effect of tail-averaging (figure 3.2). We consider averaging iterates from the start (as prescribed by [25]), after a quarter/half of total number of iterations, and unaveraged SGD as well. We see that the bias (figure 3.2a) exhibits a geometric decay in the unaveraged phase while switching to an



(a) Bias Risk (b) Variance Risk (c) Total Risk

Figure 3.2: [Zoom in to see detail] Effect of tail-averaging with mini-batch size of $b_{thresh} = 11$.

slower $\mathcal{O}(\frac{1}{t^2})$ rate with averaging. The variance (figure 3.2b) tends to increase and stabilize at $\mathcal{O}(\frac{\sigma^2}{b_{\text{thresh}}})$ in the absence of averaging, while switching to a $\mathcal{O}(\frac{1}{N})$ decay rate when averaging begins. The overall generalization error (figure 3.2c) shows the superiority of the scheme where averaging after a burn-in period allows the bias to decay towards the noise level at a geometric rate, following which tail-averaging allows us to decay the variance term, providing credence to our theoretical results regarding superiority of tail-averaged SGD.

## *3.7 Concluding Remarks*

This chapter analyzes several algorithmic primitives often used in practice in conjunction with vanilla SGD for the stochastic approximation problem. In particular, this chapter provides a sharp non-asymptotic treatment of (a) mini-batching, (b) tail-averaging, (c) effects of model mismatch, (d) behaviour of the final iterate, (e) highly parallel SGD method based on doubling batch sizes and (f) model-averaging/parameter mixing schemes for the strongly convex streaming LSR problem.

# Chapter 4

# ACCELERATING STOCHASTIC GRADIENT DESCENT (FOR LEAST SQUARES REGRESSION)

## *4.1 Chapter Notes*

This chapter presents joint work with Prateek Jain, Praneeth Netrapalli, Sham M. Kakade and Aaron Sidford and is published in the Conference on Learning Theory (COLT), 2018. This chapter's presentation is a (very minor) variant of the published version of this paper. The contributions of this chapter are summarized as follows:

- This chapter presents Accelerated SGD, which presents the first (strict) running time improvement over Stochastic Gradient Descent [99, 103, 94] for stochastic approximation problems, through considering the streaming (strongly convex) least squares regression problem, when working with sampled stochastic gradients in settings that satisfied in modern SGD implementations for Machine Learning/Artificial Intelligence setups.

- A byproduct of this result presents (the first) strict running time improvement for obtaining a solution to a potentially infinite number of consistent linear equations given access to one linear equation at a time, over the standard Kaczmarz method [58, 115].

## *4.2 Introduction*

Stochastic gradient descent (SGD) is the workhorse algorithm for optimization in machine learning and stochastic approximation problems; improving its runtime dependencies is a central issue in large scale stochastic optimization that often arise in machine learning problems at scale [12], where one can only resort to streaming algorithms.

| Algorithm | Final error | Runtime | Memory |
|---|---|---|---|
| Accelerated SVRG [2] | $\mathcal{O}\left(\frac{\sigma^2 d}{n}\right)$ | $(n+\sqrt{n\kappa})d\log\left(\frac{P(\mathbf{x}_0)-P(\mathbf{x}_*)}{(\sigma^2 d/n)}\right)$ | $nd$ |
| Streaming SVRG [35] Iterate Averaged SGD [55] | $\mathcal{O}\left(\exp\left(\frac{-n}{\kappa}\right)\cdot\left(P(\mathbf{x}_0)-P(\mathbf{x}_*)\right)+\frac{\sigma^2 d}{n}\right)$ | $nd$ | $\mathcal{O}(d)$ |
| Accelerated Stochastic Gradient Descent (this chapter) | $\mathcal{O}^*\left(\exp\left(\frac{-n}{\sqrt{\kappa\widetilde{\kappa}}}\right)\left(P(\mathbf{x}_0)-P(\mathbf{x}_*)\right)\right)+\mathcal{O}\left(\frac{\sigma^2 d}{n}\right)$ | $nd$ | $\mathcal{O}(d)$ |

Table 4.1: Comparison of this chapter's result to the best known non-asymptotic results [35, 55] for the least squares stochastic approximation problem. Here, $d, n$ are the problem dimension, number of samples; $\kappa$, $\widetilde{\kappa}$ denote the condition number and statistical condition number of the distribution; $\sigma^2$, $P(\mathbf{x}_0) - P(\mathbf{x}_*)$ denote the noise level and initial excess risk, $\mathcal{O}^*$ hides lower order terms in $d, \kappa, \widetilde{\kappa}$ (see section 4.3 for definitions and a proof for $\widetilde{\kappa} \leq \kappa$). Note that Accelerated SVRG [2] is not a streaming algorithm.

This chapter examines these broader runtime issues for the special case of stochastic approximation in the following least squares regression problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} P(\mathbf{x}), \text{ where, } P(\mathbf{x}) \overset{\text{def}}{=} \frac{1}{2}\cdot\mathbb{E}_{(\mathbf{a},b)\sim\mathcal{D}}\left[(b-\langle\mathbf{x},\mathbf{a}\rangle)^2\right], \tag{4.1}$$

where we have access to a *stochastic first order oracle*, which, when provided with $\mathbf{x}$ as an input, returns a noisy unbiased stochastic gradient using a tuple $(\mathbf{a}, b)$ sampled from $\mathcal{D}(\mathbb{R}^d \times \mathbb{R})$, with $d$ being the dimension of the problem. A query to the stochastic first-order oracle at $\mathbf{x}$ produces:

$$\widehat{\nabla}P(\mathbf{x}) = -(b-\langle\mathbf{a},\mathbf{x}\rangle)\cdot\mathbf{a}. \tag{4.2}$$

Note $\mathbb{E}\left[\widehat{\nabla}P(\mathbf{x})\right] = \nabla P(\mathbf{x})$ (i.e. eq(4.2) is an unbiased estimate). Note that nearly all practical stochastic algorithms use sampled gradients of the specific form as in equation (4.2). We discuss differences to the more general stochastic first order oracle [83] in section 4.2.4.

Let $\mathbf{x}^* \overset{\text{def}}{=} \arg\min_{\mathbf{x}} P(\mathbf{x})$ be a population risk minimizer. Given any estimation procedure which returns $\widehat{\mathbf{x}}_n$ using $n$ samples, define the *excess risk* (which we also refer to as the *generalization error* or the *error*) of $\widehat{\mathbf{x}}_n$ as $\mathbb{E}\left[P(\widehat{\mathbf{x}}_n)\right] - P(\mathbf{x}^*)$. Now, equipped a stochastic first-order oracle (equation (4.2)), our goal is to provide a computationally efficient (and streaming) estimation method whose excess risk is comparable to the optimal statistical minimax rate.

In the limit of large $n$, this minimax rate is achieved by the *empirical risk minimizer* (ERM), which is defined as follows. Given $n$ i.i.d. samples $\mathcal{S}_n = \{(\mathbf{a}_i, b_i)\}_{i=1}^n$ drawn from $\mathcal{D}$, define

$$\widehat{\mathbf{x}}_n^{\mathrm{ERM}} \overset{\mathrm{def}}{=} \arg\min_{\mathbf{x}} P_n(\mathbf{x}), \text{ where } P_n(\mathbf{x}) \overset{\mathrm{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \tfrac{1}{2}\left(b_i - \mathbf{a}_i^\top \mathbf{x}\right)^2,$$

where $\widehat{\mathbf{x}}_n^{\mathrm{ERM}}$ denotes the ERM over the samples $\mathcal{S}_n$. For the case of additive noise models (i.e. where $b = \mathbf{a}^\top \mathbf{x}^* + \epsilon$, with $\epsilon$ being independent of $\mathbf{a}$), the minimax estimation rate is $d\sigma^2/n$ [62, chap. 1,3,6], [94],[69, chap. $5-6$], [120, chap. 5,7,8], i.e.:

$$\lim_{n\to\infty} \frac{\mathbb{E}_{\mathcal{S}_n}[P(\widehat{\mathbf{x}}_n^{\mathrm{ERM}})] - P(\mathbf{x}^*)}{d\sigma^2/n} = 1, \tag{4.3}$$

where $\sigma^2 = \mathbb{E}\left[\epsilon^2\right]$ is the variance of the additive noise and the expectation is over the samples $\mathcal{S}_n$ drawn from $\mathcal{D}$. The seminal works of [103, 94] proved that a certain averaged stochastic gradient method enjoys this minimax rate, in the limit. The question we seek to address is: how fast (in a non-asymptotic sense) can we achieve the minimax rate of $d\sigma^2/n$?

### *4.2.1 Review: Acceleration with Exact Gradients*

Let us review results in convex optimization in the exact first-order oracle model. Running $t-$steps of gradient descent [18] with an exact first-order oracle yields the following guarantee:

$$P(\mathbf{x}_t) - P(\mathbf{x}^*) \le \exp\big(-t/\kappa_o\big)\cdot\big(P(\mathbf{x}_0) - P(\mathbf{x}^*)\big),$$

where $\mathbf{x}_0$ is the starting iterate, $\kappa_o = \lambda_{\max}(\mathbf{H})/\lambda_{\min}(\mathbf{H})$ is the condition number of $P(.)$, where, $\lambda_{\max}(\mathbf{H}), \lambda_{\min}(\mathbf{H})$ are the largest and smallest eigenvalue of the hessian $\mathbf{H} = \nabla^2 P(\mathbf{x}) = \mathbb{E}\left[\mathbf{a}\mathbf{a}^\top\right]$. Thus gradient descent requires $\mathcal{O}(\kappa_o)$ oracle calls to solve the problem to a given target accuracy, which is sub-optimal amongst the class of methods with access to an exact first-order oracle [86]. This sub-optimality can be addressed through Nesterov's Accelerated Gradient Descent [85], which when run for t-steps, yields the following guarantee:

$$P(\mathbf{x}_t) - P(\mathbf{x}^*) \leq \exp\big(-t/\sqrt{\kappa_o}\big) \cdot \big(P(\mathbf{x}_0) - P(\mathbf{x}^*)\big),$$

which implies that $\mathcal{O}(\sqrt{\kappa_o})$ oracle calls are sufficient to achieve a given target accuracy. This matches the oracle lower bounds [86] that state that $\Theta(\sqrt{\kappa_o})$ calls to the exact first order oracle are necessary to achieve a given target accuracy. The conjugate gradient method [48] and heavy ball method [92] are also known to obtain this convergence rate for solving a system of linear equations and for quadratic functions. These methods are termed fast gradient methods owing to the improvements offered by these methods over Gradient Descent. This chapter seeks to address the question: "Can we accelerate stochastic approximation in a manner similar to what has been achieved with the exact first order oracle model?"

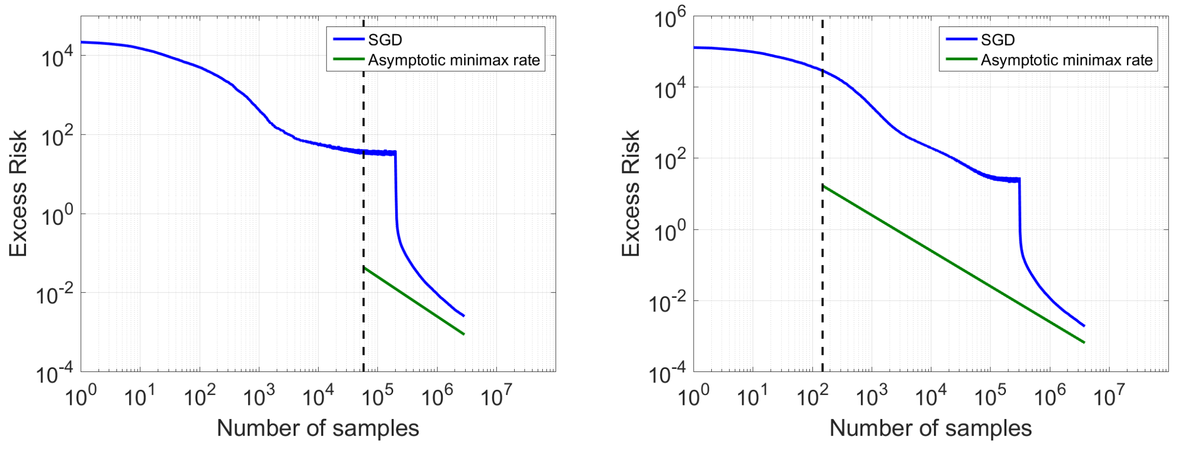
### *4.2.2 A thought experiment: Is Accelerating Stochastic Approximation possible?*

Let us recollect known results in stochastic approximation for the least squares regression problem (in equation (4.1)). Running $n$-steps of tail-averaged SGD [55] (or, streaming SVRG [35][1]) provides an output $\widehat{\mathbf{x}}_n$ that satisfies the following excess risk bound:

$$\mathbb{E}\left[P(\widehat{\mathbf{x}}_n)\right] - P(\mathbf{x}^*) \leq \exp(-n/\kappa) \cdot \big(P(\mathbf{x}_0) - P(\mathbf{x}^*)\big) + 2\sigma^2 d/n, \tag{4.4}$$

where $\kappa$ is the condition number of the distribution, which can be upper bounded as $L/\lambda_{\min}(\mathbf{H})$, assuming that $\|\mathbf{a}\| \leq L$ with probability one (refer to section 4.3 for a precise definition of $\kappa$). Under appropriate assumptions, these are the best known rates under the stochastic first order oracle model (see section 4.2.4 for further discussion). A natural implication of the bound implied by averaged SGD is that with $\widetilde{\mathcal{O}}(\kappa)$ oracle calls [55] (where, $\widetilde{\mathcal{O}}(\cdot)$ hides log factors in $d, \kappa$), the excess risk attains (up to constants) the (asymptotic) minimax statistical rate. Note that the excess risk bounds in stochastic approximation consist of two terms: (a) *bias*: which represents the dependence of the generalization error on the

[1]Streaming SVRG does not function in the stochastic first order oracle model [35]

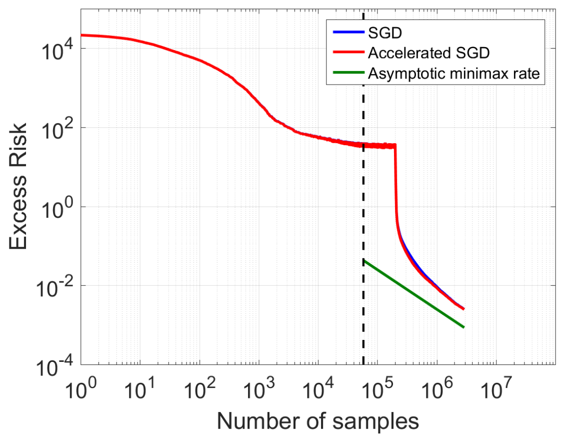


(a) Discrete distribution (b) Gaussian distribution

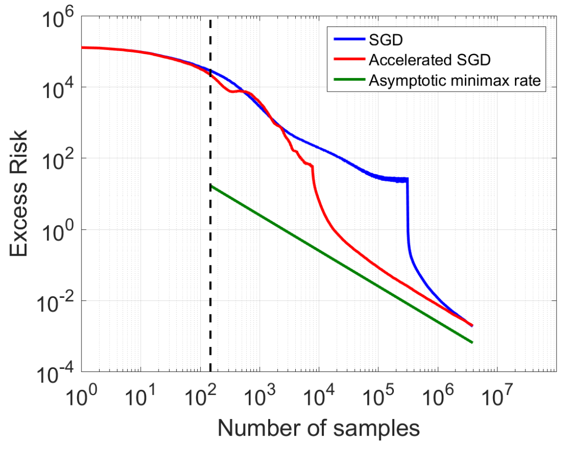
Figure 4.1: Plot of error vs number of samples for averaged SGD and the minimax risk for the discrete and Gaussian distributions with $d = 50$, $\kappa \approx 10^5$ (see section 4.2.2 for details on the distribution). The kink in the SGD curve represents when the tail-averaging phase begins [55]; this point is chosen appropriately. The green curves show the asymptotically optimal minimax rate of $d\sigma^2/n$. The vertical dashed line shows the sample size at which the empirical covariance, $\frac{1}{n}\sum_{i=1}^{n}\mathbf{a}_i\mathbf{a}_i^\top$, becomes full rank, which is shown at $\frac{1}{\min_i p_i}$ in the discrete case and $d$ in the Gaussian case. With fewer samples than this (i.e. before the dashed line), it is information theoretically not possible to guarantee non-trivial risk (without further assumptions). For the Gaussian case, note how the behavior of SGD is far from the dotted line; it is this behavior that one might hope to improve upon. See the text for a discussion.

initial excess risk $P(\mathbf{x}_0) - P(\mathbf{x}^*)$, and (b) the *variance:* which represents the dependence of the generalization error on the noise level $\sigma^2$ in the problem.

A precise question regarding accelerating stochastic approximation is: "is it possible to improve the rate of decay of the bias term, while retaining (up to constants) the statistical minimax rate?" The key technical challenge in answering this question is in sharply characterizing the error accumulation of fast gradient methods in the stochastic approximation setting. Common folklore and prior work suggest otherwise: several efforts have attempted



(a) Discrete distribution



(b) Gaussian distribution

Figure 4.2: Plot of total error vs number of samples for averaged SGD, (this chapter's) accelerated SGD method and the minimax risk for the discrete and Gaussian distributions with $d = 50, \kappa \approx 10^5$ (see section 4.2.2 for details). For the discrete case, accelerated SGD degenerates to SGD, which nearly matches the minimax risk (when it becomes well defined). For the Gaussian case, accelerated SGD significantly improves upon SGD.

to quantify instabilities in the face of statistical or non-statistical errors [91, 95, 93, 42, 102, 113, 22, 27, 28, 125]. Refer to section 4.2.4 for a discussion on robustness of acceleration to error accumulation. Optimistically, as suggested by the gains enjoyed by accelerated methods in the exact first order oracle model, we may hope to replace the $\widetilde{\mathcal{O}}(\kappa)$ oracle calls achieved by averaged SGD to $\widetilde{\mathcal{O}}(\sqrt{\kappa})$. We now provide a counter example, showing that such an improvement is not possible. Consider a (discrete) distribution $\mathcal{D}$ where the input $\mathbf{a}$ is the $i^{\text{th}}$ standard basis vector with probability $p_i$, $\forall\ i = 1, 2, ..., d$. The covariance of $\mathbf{a}$ in this case is a diagonal matrix with diagonal entries $p_i$. The condition number of this distribution is $\kappa = \frac{1}{\min_i p_i}$. In this case, it is impossible to make non-trivial reduction in error by observing fewer than $\kappa$ samples, since with constant probability, we would not have seen the vector corresponding to the smallest probability.

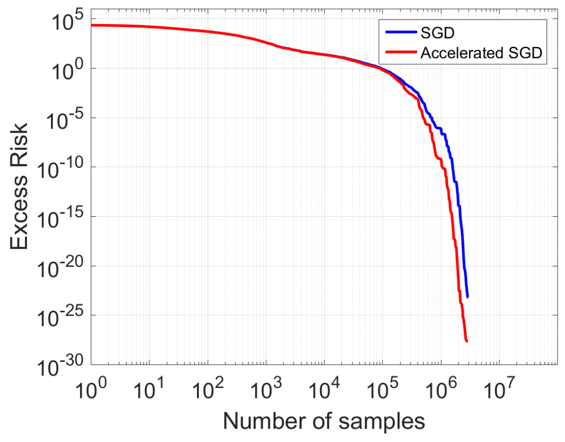
On the other hand, consider a case where the distribution $\mathcal{D}$ is a Gaussian with a large condition number $\kappa$. Matrix concentration informs us that (with high probability and irrespective of how large $\kappa$ is) after observing $n = \mathcal{O}(d)$ samples, the empirical covariance matrix will be a spectral approximation to the true covariance matrix, i.e. for some constant $c > 1$, $\mathbf{H}/c \preceq \frac{1}{n}\sum_{i=1}^{n} \mathbf{a}_i\mathbf{a}_i^\top \preceq c\mathbf{H}$. Here, we may hope to achieve a faster convergence rate, as information theoretically $\mathcal{O}(d)$ samples suffice to obtain a non-trivial statistical estimate (see [50] for further discussion).

Figure 4.1 shows the behavior of SGD in these cases; both are synthetic examples in 50−dimensions, with a condition number $\kappa \approx 10^5$ and noise level $\sigma^2 = 100$. See the figure caption for more details.

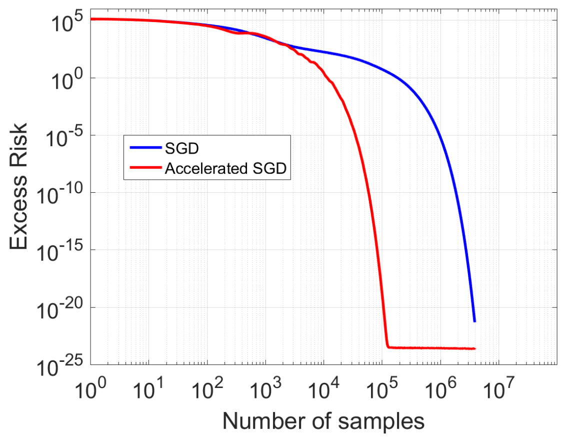
These examples suggest that if acceleration is indeed possible, then the degree of improvement (say, over averaged SGD) must depend on distributional quantities that go beyond the condition number $\kappa$. A natural conjecture is that this improvement must depend on the number of samples required to spectrally approximate the covariance matrix of the distribution; below this sample size it is not possible to obtain any non-trivial statistical estimate due to information theoretic reasons. This sample size is quantified by a notion which we refer to as the *statistical condition number* $\widetilde{\kappa}$ (see section 4.3 for a precise definition and for further discussion about $\widetilde{\kappa}$). As we will see in section 4.3, we have $\widetilde{\kappa} \leq \kappa$, $\widetilde{\kappa}$ is affine invariant, unlike $\kappa$ (i.e. $\widetilde{\kappa}$ is invariant to linear transformations over $\mathbf{a}$).

### *4.2.3 Contributions*

This chapter introduces an accelerated stochastic gradient descent scheme, which can be viewed as a stochastic variant of Nesterov's accelerated gradient method [87]. As pointed out in Section 4.2.2, the excess risk of this algorithm can be decomposed into two parts namely, *bias* and *variance*. For the stochastic approximation problem of least squares regression, this chapter establishes bias contraction at a geometric rate of $\mathcal{O}(1/\sqrt{\kappa\widetilde{\kappa}})$, improving over prior results [35, 55],which prove a geometric rate of $\mathcal{O}(1/\kappa)$, while retaining statistical minimax rates (up to constants) for the variance. Here $\kappa$ is the condition number and $\widetilde{\kappa}$ is the statistical condition number of the distribution, and a rate of $\mathcal{O}(1/\sqrt{\kappa\widetilde{\kappa}})$ is an improvement



(a) Discrete distribution



(b) Gaussian distribution

Figure 4.3: Comparison of averaged SGD with this chapter's accelerated SGD in the absence of noise ($\sigma^2 = 0$) for the Gaussian and Discrete distribution with $d = 50, \kappa \approx 10^5$. Acceleration yields substantial gains over averaged SGD for the Gaussian case, while degenerating to SGD's behavior for the discrete case. See section 4.2.2 for discussion.

over $\mathcal{O}(1/\kappa)$ since $\widetilde{\kappa} \leq \kappa$ (see Section 4.3 for definitions and a short proof of $\widetilde{\kappa} \leq \kappa$).

See Table 4.1 for a theoretical comparison. Figure 4.2 provides an empirical comparison of the proposed (tail-averaged) accelerated algorithm to (tail-averaged) SGD [55] on our two running examples. Our result presents the first improvement over SGD even in the noiseless (i.e. realizable) case where $\sigma = 0$; this case is equivalent to the setting where we have a distribution over a (possibly infinite) set of consistent linear equations [58, 115]. See Figure 4.3 for a comparison on the case where $\sigma = 0$.

On a more technical note, this chapter introduces two new techniques in order to analyze the proposed accelerated stochastic gradient method: (a) the chapter introduces a new potential function in order to show faster rates of decaying the bias, and (b) the chapter provides a sharp understanding of the behavior of the proposed accelerated stochastic gradient descent updates as a stochastic process and utilizes this in providing a near-exact estimate of the covariance of its iterates. This viewpoint is critical in order to prove that the algorithm achieves the statistical minimax rate.

We use the operator viewpoint for analyzing stochastic gradient methods, introduced in [25]. This viewpoint was also used in [29, 55].

### *4.2.4 Related Work*

**Non-asymptotic Stochastic Approximation:** Stochastic gradient descent (SGD) and its variants are by far the most widely studied algorithms for the stochastic approximation problem. While initial works [99] considered the final iterate of SGD, later works [103, 94] demonstrated that averaged SGD obtains statistically optimal estimation rates. Several works provide non-asymptotic analyses for averaged SGD and variants [5, 6, 35] for various stochastic approximation problems. For stochastic approximation with least squares regression [7, 25, 81, 35, 55] provide non-asymptotic analysis of the behavior of SGD and its variants. [25, 29] provide non-asymptotic results which achieve the minimax rate on the variance (where the bias is lower order, not geometric). [81] achieves a geometric rate on the bias (and where the variance is not minimax). [35, 55] obtain both the minimax rate on the variance and a geometric rate on the bias, as seen in equation (4.4).

**Acceleration and Noise Stability:** While there have been several attempts at understanding if it is possible to accelerate SGD , the results have been largely negative. With regards to acceleration with adversarial (non-statistical) errors in the exact first order oracle model, [22] provide negative results and [27, 28] provide lower bounds showing that fast gradient methods do not improve upon standard gradient methods. There is also a series of works considering statistical errors. [93] suggests that the relative merits of heavy ball (HB) method [92] in the noiseless case vanish with noise unless strong assumptions on the noise model are considered; an instance of this is when the noise variance decays as the iterates approach the minimizer. The Conjugate Gradient (CG) method [48] is suggested to face similar robustness issues in the face of statistical errors [93]; this is in addition to the issues that CG is known to suffer from owing to roundoff errors (due to finite precision arithmetic) [91, 42]. In the signal processing literature, SGD goes by the name of Least Mean Squares (LMS) [122, chap. $5-9$] algorithm. Paralleling notions of minimax optimality (in the MLE sense), the LMS algorithm is robust (and optimal) in a $H^\infty$ sense [44], i.e., small disturbances (*i.e.* noise) and modeling errors (defined in an appropriate sense) provably yields small estimation errors. To our knowledge, we aren't aware of similar notions studied in the context of fast (*i.e.* accelerated) LMS methods. That said, in the LMS literature, there have been efforts that date to several decades [95, 102, 113] which study accelerated LMS methods (stochastic variants of CG/HB) in the same oracle model as the one considered by this chapter (equation 4.2). These efforts consider the final iterate (i.e. no iterate averaging) of accelerated LMS methods with a fixed step-size and conclude that while it allows for a faster decay of the initial error (bias) (which is unquantified), their steady state behavior (i.e. variance) is worse compared to that of LMS. [125] considered a constant step size accelerated scheme with no iterate averaging in the same oracle model as this chapter, and conclude that these do not offer any improvement over standard SGD. More concretely, [125] show that the variance of their accelerated SGD method with a sufficiently small constant step size is the same as that of SGD with a significantly larger step size. Note that none of the these efforts [95, 102, 113, 125] achieve minimax error rates or quantify (any improvement whatsoever on the) rate of bias decay.

**Oracle models and optimality:** With regards to notions of optimality, there are (at least) two lines of thought: one is a statistical objective where the goal is (on every problem instance) to match the rate of the statistically optimal estimator [4, 33, 94], [62, chap. 1,3,6]; another is on obtaining algorithms whose worst case upper bounds (under various assumptions such as bounded noise) match the lower bounds provided in [83]. The work of [94] are in the former model, where they show that the distribution of the averaged SGD estimator matches, on *every* problem, that of the statistically optimal estimator, in the limit (under appropriate regularization conditions standard in the statistics literature, where the optimal estimator is often referred to as the maximum likelihood estimator/the empirical risk minimizer/an $M$-estimator [69, chap. 5 − 6], [120, chap. 5,7,8]). Along these lines, non-asymptotic rates towards statistically optimal estimators are given by [7, 25, 29, 35, 55]. This chapter can be seen as improving this non-asymptotic rate (to the statistically optimal estimation rate) using an accelerated method. As to the latter (i.e. matching the worst-case lower bounds in [83]), there are a number of positive results on using accelerated stochastic optimization procedures; the works of [67, 51, 38, 39, 30] match the lower bounds provided in [83]. We compare these assumptions and works in more detail.

In stochastic first order oracle models (see [62, chap. 1, 3, 6], [64, chap. 5,10,11]), one typically has access to sampled gradients of the form:

$$\widehat{\nabla} P(\mathbf{x}) = \nabla P(\mathbf{x}) + \boldsymbol{\eta}, \tag{4.5}$$

where varying assumptions are made on the noise $\boldsymbol{\eta}$. The worst-case lower bounds in [83] are based on that $\boldsymbol{\eta}$ is bounded; the accelerated methods in [67, 51, 38, 39, 30] which match these lower bounds in various cases, all assume either bounded noise or, at least $\mathbb{E}\left[\|\boldsymbol{\eta}\|^2\right]$ is finite. In the least squares setting (such as the one often considered in practice and also considered in [94, 7, 25, 29, 35, 55]), this assumption does not hold, since $\mathbb{E}\left[\|\boldsymbol{\eta}\|^2\right]$ is not bounded. To see this, $\boldsymbol{\eta}$ in our oracle model (equation (4.2)) is:

$$\boldsymbol{\eta} = \widehat{\nabla} P(\mathbf{x}) - \nabla P(\mathbf{x}) = (\mathbf{a}\mathbf{a}^\top - \mathbf{H})(\mathbf{x} - \mathbf{x}^*) - \epsilon \cdot \mathbf{a} \tag{4.6}$$

which implies that $\mathbb{E}\left[\|\boldsymbol{\eta}\|^2\right]$ is not uniformly bounded (unless additional assumptions are enforced to ensure that the algorithm's iterates $\mathbf{x}$ lie within a compact set). Hence, the assumptions made in [51, 38, 39, 30] do not permit one to obtain finite $n$-sample bounds on the excess risk. Suppose we consider the case of $\epsilon = 0$, i.e. where the additive noise is zero and $b = \mathbf{a}^\top \mathbf{x}^*$. For this case, this chapter provides a geometric rate of convergence to the minimizer $\mathbf{x}^*$, while the results of [38, 39, 30] at best indicate a $\mathcal{O}(1/n)$ rate. Finally, in contrast to all other existing work, our result is the first to provide finer distribution dependent characteristics of the improvements offered by accelerating SGD (e.g. refer to the Gaussian and discrete examples in section 4.2.2).

**Acceleration and Finite Sums:** As a final remark, there have been results [109, 34, 72, 68, 2] that provide accelerated rates for *offline* stochastic optimization which deal with minimizing sums of convex functions; these results are almost tight due to matching lower bounds [68, 124]. These results do not immediately translate into rates on the generalization error. Furthermore, these algorithms are not streaming, as they require making multiple passes over a dataset stored in memory. Refer to [35] for more details.

## 4.3 Main Results

We now provide our assumptions and main result, before which, we have some notation. For a vector $\mathbf{x} \in \mathbb{R}^d$ and a positive semi-definite matrix $\mathbf{S} \in \mathbb{R}^{d\times d}$ (i.e. $\mathbf{S} \succeq 0$), denote $\|\mathbf{x}\|_{\mathbf{S}}^2 \overset{\text{def}}{=} \mathbf{x}^\top \mathbf{S}\mathbf{x}$.

### 4.3.1 Assumptions and Definitions

Let $\mathbf{H}$ denote the second moment matrix of the input, which is also the hessian $\nabla^2 P(\mathbf{x})$ of (4.1):

$$\mathbf{H} \overset{\text{def}}{=} \mathbb{E}_{(\mathbf{a},b)\sim\mathcal{D}}\left[\mathbf{a}\otimes\mathbf{a}\right] = \nabla^2 P(\mathbf{x}).$$

Furthermore, let the fourth moment tensor $\mathcal{M}$ of the inputs $\mathbf{a}\sim\mathcal{D}$ is defined as:

$$\mathcal{M} = \mathbb{E}_{(\mathbf{a},b)\sim\mathcal{D}}\left[\mathbf{a}\otimes\mathbf{a}\otimes\mathbf{a}\otimes\mathbf{a}\right].$$

($\mathcal{A}$**1**) **Finite second and fourth moment:** The second moment matrix $\mathbf{H}$ and the fourth moment tensor $\mathcal{M}$ exist and are finite.

($\mathcal{A}$**2**) **Positive Definiteness**: The second moment matrix $\mathbf{H}$ is strictly positive definite, i.e. $\mathbf{H} \succ 0$.

We assume ($\mathcal{A}$**1**) and ($\mathcal{A}$**2**). ($\mathcal{A}$**2**) implies that $P(\mathbf{x})$ is *strongly convex* and admits a unique minimizer $\mathbf{x}^*$. Denote the noise $\epsilon$ in a sample $(\mathbf{a},b)\sim\mathcal{D}$ as: $\epsilon \overset{\text{def}}{=} b - \langle \mathbf{a},\mathbf{x}^*\rangle$. First order optimality conditions of $\mathbf{x}^*$ imply

$$\nabla P(\mathbf{x}^*) = \mathbb{E}\left[\epsilon\cdot\mathbf{a}\right] = 0.$$

Let $\mathbf{\Sigma}$ denote the covariance of gradient at optimum $\mathbf{x}^*$ (or *noise covariance matrix*),

$$\mathbf{\Sigma} \overset{\text{def}}{=} \mathbb{E}_{(\mathbf{a},b)\sim\mathcal{D}}\left[\widehat{\nabla}P(\mathbf{x}^*) \otimes \widehat{\nabla}P(\mathbf{x}^*)\right] = \mathbb{E}_{(\mathbf{a},b)\sim\mathcal{D}}\left[\epsilon^2 \cdot \mathbf{a} \otimes \mathbf{a}\right].$$

We define the *noise level* $\sigma^2$, *condition number* $\kappa$, *statistical condition number* $\widetilde{\kappa}$ below.

**Noise level**: The *noise level* is defined to be the smallest positive number $\sigma^2$ such that

$$\mathbf{\Sigma} \preceq \sigma^2 \mathbf{H}.$$

The noise level $\sigma^2$ quantifies the amount of noise in the stochastic gradient oracle and has been utilized in previous work (e.g., see [5, 7]) for providing non-asymptotic bounds for the stochastic approximation problem. In the *homoscedastic* (additive noise/well specified) case, where $\epsilon$ is independent of the input $\mathbf{a}$, this condition is satisfied with equality, i.e. $\mathbf{\Sigma} = \sigma^2\, \mathbf{H}$ with $\sigma^2 = \mathbb{E}\left[\epsilon^2\right]$.

**Condition number**: Let

$$\mu \overset{\text{def}}{=} \lambda_{\min}(\mathbf{H}).$$

$\mu > 0$ by ($\mathcal{A}\mathbf{2}$). Now, let $R^2$ be the smallest positive number such that

$$\mathbb{E}\left[\|\mathbf{a}\|^2\, \mathbf{a}\mathbf{a}^\top\right] \preceq R^2\, \mathbf{H}.$$

. The *condition number* $\kappa$ of the distribution $\mathcal{D}$ [25, 55] is

$$\kappa \overset{\text{def}}{=} R^2/\mu.$$

**Statistical condition number**: The *statistical condition number* $\widetilde{\kappa}$ is defined as the smallest positive number such that

$$\mathbb{E}\left[\|\mathbf{a}\|^2_{\mathbf{H}^{-1}}\, \mathbf{a}\mathbf{a}^\top\right] \preceq \widetilde{\kappa}\, \mathbf{H}.$$

**Remarks on $\widetilde{\kappa}$ and $\kappa$**: Unlike $\kappa$, it is straightforward to see that $\widetilde{\kappa}$ is affine invariant (i.e. unchanged with linear transformations over $\mathbf{a}$). Since $\mathbb{E}\left[\|\mathbf{a}\|^2_{\mathbf{H}^{-1}}\, \mathbf{a}\mathbf{a}^\top\right] \preceq \frac{1}{\mu}\mathbb{E}\left[\|\mathbf{a}\|_2^2\, \mathbf{a}\mathbf{a}^\top\right] \preceq \kappa\mathbf{H}$, we note $\widetilde{\kappa} \leq \kappa$. For the discrete case (from Section 4.2.2), it is straightforward to see that both $\kappa$ and $\widetilde{\kappa}$ are equal to $1/\min_i p_i$. In contrast, for the Gaussian case (from Section 4.2.2),

**Algorithm 3** (Tail-Averaged) **A**ccelerated **S**tochastic **G**radient **D**escent (ASGD)

```
Require: $n$ oracle calls (4.2), initial point $\mathbf{x}_0 = \mathbf{v}_0$, Unaveraged (burn-in) phase $t$, Step size
    parameters $\alpha, \beta, \gamma, \delta$
1: for $j = 1, \cdots n$ do
2:     $\mathbf{y}_{j-1} \leftarrow \alpha \mathbf{x}_{j-1} + (1-\alpha)\mathbf{v}_{j-1}$
3:     $\mathbf{x}_j \leftarrow \mathbf{y}_{j-1} - \delta \widehat{\nabla} P(\mathbf{y}_{j-1})$
4:     $\mathbf{z}_{j-1} \leftarrow \beta \mathbf{y}_{j-1} + (1-\beta)\mathbf{v}_{j-1}$
5:     $\mathbf{v}_j \leftarrow \mathbf{z}_{j-1} - \gamma \widehat{\nabla} P(\mathbf{y}_{j-1})$
6: end for
Ensure: $\bar{\mathbf{x}}_{t,n} \leftarrow \frac{1}{n-t}\sum_{j=t+1}^{n} \mathbf{x}_j$
```

$\widetilde{\kappa}$ is $\mathcal{O}(d)$, while $\kappa$ is $\mathcal{O}(\text{Trace}(\mathbf{H})/\mu)$ which may be arbitrarily large (based on choice of the coordinate system).

$\widetilde{\kappa}$ governs how many samples $\mathbf{a}_i$ require to be drawn from $\mathcal{D}$ so that the empirical covariance is spectrally close to $\mathbf{H}$, i.e. for some constant $c > 1$, $\mathbf{H}/c \preceq \frac{1}{n}\sum_{i=1}^{n} \mathbf{a}_i \mathbf{a}_i^\top \preceq c\mathbf{H}$. In comparison to the matrix Bernstein inequality where stronger (yet related) moment conditions are assumed in order to obtain high probability results, our results hold only in expectation (refer to [50] for this definition, wherein $\widetilde{\kappa}$ is referred to as bounded statistical leverage in Theorem 1 and remark 1).

### *4.3.2 Algorithm and Main Theorem*

Algorithm 3 presents the pseudo code of the proposed algorithm. ASGD can be viewed as a variant of Nesterov's accelerated gradient method [87], working with a stochastic gradient oracle (equation (4.2)) and with tail-averaging the final $n-t$ iterates. The main result now follows:

**Theorem 9.** *Suppose* ($\mathcal{A}\mathbf{1}$) *and* ($\mathcal{A}\mathbf{2}$) *hold. Set* $\alpha = \frac{3\sqrt{5}\cdot\sqrt{\kappa\widetilde{\kappa}}}{1+3\sqrt{5}\cdot\sqrt{\kappa\widetilde{\kappa}}}, \beta = \frac{1}{9\sqrt{\kappa\widetilde{\kappa}}}, \gamma = \frac{1}{3\sqrt{5}\cdot\mu\sqrt{\kappa\widetilde{\kappa}}}, \delta = \frac{1}{5R^2}$. *After $n$ calls to the stochastic first order oracle (equation (4.2)), ASGD outputs* $\bar{\mathbf{x}}_{t,n}$

*satisfying:*

$$\mathbb{E}\left[P(\bar{\mathbf{x}}_{t,n})\right] - P(\mathbf{x}^*) \leq \underbrace{C \cdot \frac{(\kappa\widetilde{\kappa})^{9/4} d\kappa}{(n-t)^2} \cdot \exp\left(\frac{-t}{9\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \big(P(\mathbf{x}_0) - P(\mathbf{x}^*)\big)}_{\text{Leading order bias error}} + \underbrace{5\frac{\sigma^2 d}{n-t}}_{\text{Leading order variance}}$$

$$+ \underbrace{C \cdot (\kappa\widetilde{\kappa})^{5/4} d\kappa \cdot \exp\left(\frac{-n}{9\sqrt{\kappa\widetilde{\kappa}}}\right)\big(P(\mathbf{x}_0) - P(\mathbf{x}^*)\big)}_{\text{Exponentially vanishing lower order bias term}} + \underbrace{C \cdot \frac{\sigma^2 d}{(n-t)^2}\sqrt{\kappa\widetilde{\kappa}}}_{\text{Lower order variance error term}} +$$

$$\underbrace{C \cdot \exp\left(-\frac{n}{9\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \left(\sigma^2 d \cdot (\kappa\widetilde{\kappa})^{7/4} + \frac{\sigma^2 d}{(n-t)^2} \cdot (\kappa\widetilde{\kappa})^{7/2}\widetilde{\kappa}\right)}_{\text{Exponentially vanishing lower order variance error terms}}$$

$$+ \underbrace{C \cdot \frac{\sigma^2 d}{n-t}(\kappa\widetilde{\kappa})^{11/4} \exp\left(-\frac{(n-t-1)}{30\sqrt{\kappa\widetilde{\kappa}}}\right)}_{\text{Exponentially vanishing lower order variance error term}},$$

*where $C$ is a universal constant, $\sigma^2$, $\kappa$ and $\widetilde{\kappa}$ are the noise level, condition number and statistical condition number respectively.*

The following corollary holds if the iterates are tail-averaged over the last $n/2$ samples and $n > \mathcal{O}(\sqrt{\kappa\widetilde{\kappa}}\log(d\kappa\widetilde{\kappa}))$. The second condition lets us absorb lower order terms into leading order terms.

**Corollary 10.** *Assume the parameter settings of Theorem 9 and let $t = \lfloor n/2 \rfloor$ and $n > C'\sqrt{\kappa\widetilde{\kappa}}\log(d\kappa\widetilde{\kappa})$ (for an appropriate universal constants $C, C'$). We have that with $n$ calls to the stochastic first order oracle, ASGD outputs a vector $\bar{\mathbf{x}}_{t,n}$ satisfying:*

$$\mathbb{E}\left[P(\bar{\mathbf{x}}_{t,n})\right] - P(\mathbf{x}^*) \leq C \cdot \exp\left(-\frac{n}{20\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \big(P(\mathbf{x}_0) - P(\mathbf{x}^*)\big) + 11\frac{\sigma^2 d}{n}.$$

A few remarks about the result of Theorem 9 are due: (i) ASGD decays the initial error at a geometric rate of $\mathcal{O}(1/\sqrt{\kappa\widetilde{\kappa}})$ during the unaveraged phase of $t$ iterations, which presents the first improvement over the $\mathcal{O}(1/\kappa)$ rate offered by SGD [99]/averaged SGD [94, 55] for the least squares stochastic approximation problem, (ii) the second term in the error bound indicates that ASGD obtains (up to constants) the minimax rate once $n > \mathcal{O}(\sqrt{\kappa\widetilde{\kappa}}\log(d\kappa\widetilde{\kappa}))$. Note that this implies that Theorem 9 provides a sharp non-asymptotic analysis (up to log factors) of the behavior of Algorithm 3.

### *4.3.3 Discussion and Open Problems*

A challenging problem in this context is in formalizing a finite sample size lower bound in the oracle model considered in this chapter. Lower bounds in stochastic oracle models have been considered in the literature (see [83, 96, 1]), though it is not evident these oracle models and lower bounds are sharp enough to imply statements in our setting (see section 4.2.4 for a discussion of these oracles).

Let us now understand Theorem 9 in the broader context of stochastic approximation. Under certain regularity conditions, it is known that [69, chap. $5-6$], [120, chap. 5,7,8] that the rate described in equation (4.3) for the homoscedastic case holds for a broader set of misspecified models (i.e., heteroscedastic noise case), with an appropriate definition of the noise variance. By defining $\sigma^2_{\text{ERM}} \overset{\text{def}}{=} \mathbb{E}\left[\left\|\widehat{\nabla} P(\mathbf{x}^*)\right\|^2_{\mathbf{H}^{-1}}\right]$, the rate of the ERM is guaranteed to approach $\sigma^2_{\text{ERM}}/n$ [69, 120] in the limit of large $n$, i.e.:

$$\lim_{n\to\infty} \frac{\mathbb{E}_{\mathcal{S}_n}[P_n(\widehat{\mathbf{x}}_n^{\text{ERM}})] - P(\mathbf{x}^*)}{\sigma^2_{\text{ERM}}/n} = 1, \tag{4.7}$$

where $\widehat{\mathbf{x}}_n^{\text{ERM}}$ is the ERM over samples $\mathcal{S}_n = \{\mathbf{a}_i, b_i\}_{i=1}^n$. Averaged SGD [55] and streaming SVRG [35] are known to achieve these rates for the heteroscedastic case. Refer to [35] for more details.Neglecting constants, Theorem 9 is guaranteed to achieve the rate of the ERM for the *homoscedastic* case (where $\mathbf{\Sigma} = \sigma^2\mathbf{H}$) and is tight when the bound $\mathbf{\Sigma} \preceq \sigma^2\mathbf{H}$ is nearly tight (upto constants). We conjecture ASGD achieves the rate of the ERM in the heteroscedastic case by appealing to a more refined analysis as is the case for averaged SGD (see [55]). It is also an open question to understand acceleration for smooth stochastic approximation (beyond least squares), in situations where the rate represented by equation (4.7) holds [94].

## ***4.4 Proof Outline***

We now present a brief outline of the proof of Theorem 9. Recall the variables in Algorithm 3. Before presenting the proof outline we require some definitions. We begin by defining the centered estimate $\boldsymbol{\theta}_j$ as:

$$\boldsymbol{\theta}_j \overset{\text{def}}{=} \begin{bmatrix} \mathbf{x}_j - \mathbf{x}^* \\ \mathbf{y}_j - \mathbf{x}^* \end{bmatrix} \in \mathbb{R}^{2d}.$$

Recall that the stepsizes in Algorithm 3 are $\alpha = \frac{3\sqrt{5}\cdot\sqrt{\kappa\tilde{\kappa}}}{1+3\sqrt{5}\cdot\sqrt{\kappa\tilde{\kappa}}}, \beta = \frac{1}{9\sqrt{\kappa\tilde{\kappa}}}, \gamma = \frac{1}{3\sqrt{5}\cdot\mu\sqrt{\kappa\tilde{\kappa}}}, \delta = \frac{1}{5R^2}$. The accelerated SGD updates of Algorithm 3 can be written in terms of $\boldsymbol{\theta}_j$ as:

$$\boldsymbol{\theta}_j = \widehat{\mathbf{A}}_j\boldsymbol{\theta}_{j-1} + \boldsymbol{\zeta}_j, \quad \text{where,}$$

$$\widehat{\mathbf{A}}_j \overset{\text{def}}{=} \begin{bmatrix} 0 & (\mathbf{I} - \delta\mathbf{a}_j\mathbf{a}_j^\top) \\ -\alpha(1-\beta)\ \mathbf{I} & (1+\alpha(1-\beta))\mathbf{I} - (\alpha\delta + (1-\alpha)\gamma)\mathbf{a}_j\mathbf{a}_j^\top \end{bmatrix}, \boldsymbol{\zeta}_j \overset{\text{def}}{=} \begin{bmatrix} \delta\cdot\epsilon_j\mathbf{a}_j \\ (\alpha\delta + (1-\alpha)\gamma)\cdot\epsilon_j\mathbf{a}_j \end{bmatrix},$$

where $\epsilon_j = b_j - \langle \mathbf{a}_j, \mathbf{x}^*\rangle$. The tail-averaged iterate $\bar{\mathbf{x}}_{t,n}$ is associated with its own centered estimate $\bar{\boldsymbol{\theta}}_{t,n} \overset{\text{def}}{=} \frac{1}{n-t}\sum_{j=t+1}^{n}\boldsymbol{\theta}_j$. Let $\mathbf{A} \overset{\text{def}}{=} \mathbb{E}\left[\widehat{\mathbf{A}}_j|\mathcal{F}_{j-1}\right]$, where $\mathcal{F}_{j-1}$ is a filtration generated by $(\mathbf{a}_1, b_1), \cdots, (\mathbf{a}_{j-1}, b_{j-1})$. Let $\mathcal{B}, \mathcal{A}_{\mathcal{L}}, \mathcal{A}_{\mathcal{R}}$ be linear operators acting on a matrix $\mathbf{S} \in \mathbb{R}^{2d\times 2d}$ so that $\mathcal{B}\mathbf{S} \overset{\text{def}}{=} \mathbb{E}\left[\widehat{\mathbf{A}}_j\mathbf{S}\widehat{\mathbf{A}}_j^\top|\mathcal{F}_{j-1}\right]$, $\mathcal{A}_{\mathcal{L}}\mathbf{S} \overset{\text{def}}{=} \mathbf{A}\mathbf{S}$, $\mathcal{A}_{\mathcal{R}}\mathbf{S} \overset{\text{def}}{=} \mathbf{S}\mathbf{A}$. Denote $\widehat{\boldsymbol{\Sigma}} \overset{\text{def}}{=} \mathbb{E}\left[\boldsymbol{\zeta}_j\boldsymbol{\zeta}_j^\top|\mathcal{F}_{j-1}\right]$ and matrices $\mathbf{G}, \mathbf{Z}, \widetilde{\mathbf{G}}$ as:

$$\mathbf{G} \overset{\text{def}}{=} \widetilde{\mathbf{G}}^\top\mathbf{Z}\widetilde{\mathbf{G}}, \text{where, } \widetilde{\mathbf{G}} \overset{\text{def}}{=} \begin{bmatrix} \mathbf{I} & 0 \\ \frac{-\alpha}{1-\alpha}\mathbf{I} & \frac{1}{1-\alpha}\mathbf{I} \end{bmatrix}, \quad \mathbf{Z} \overset{\text{def}}{=} \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mu\mathbf{H}^{-1} \end{bmatrix}.$$

**Bias-variance decomposition**: The proof of Theorem 9 employs the *bias-variance* decomposition, which is well known in the context of stochastic approximation (see [5, 35, 55]) and is re-derived in the appendix. The bias-variance decomposition allows for the generalization error to be upper-bounded by analyzing two sub-problems: (a) *bias*, analyzing the algorithm's behavior on the *noiseless* problem (i.e. $\boldsymbol{\zeta}_j = 0\ \forall\ j$ a.s.) while starting at $\boldsymbol{\theta}_0^{\text{bias}} = \boldsymbol{\theta}_0$ and (b) *variance*, analyzing the algorithm's behavior by starting at the solution (i.e. $\boldsymbol{\theta}_0^{\text{variance}} = 0$) and allowing the noise $\boldsymbol{\zeta}_\cdot$ to drive the process. In a similar manner as $\bar{\boldsymbol{\theta}}_{t,n}$, the bias and variance sub-problems are associated with $\bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}}$ and $\bar{\boldsymbol{\theta}}_{t,n}^{\text{variance}}$, and these are related as:

$$\mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n} \otimes \bar{\boldsymbol{\theta}}_{t,n}\right] \preceq 2\cdot\left(\mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}} \otimes \bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}}\right] + \mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n}^{\text{variance}} \otimes \bar{\boldsymbol{\theta}}_{t,n}^{\text{variance}}\right]\right). \tag{4.8}$$

Since we deal with the square loss, the generalization error of the output $\bar{\mathbf{x}}_{t,n}$ of Algorithm 3 is:

$$\mathbb{E}\left[P(\bar{\mathbf{x}}_{t,n})\right] - P(\mathbf{x}^*) = \frac{1}{2} \cdot \langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n} \otimes \bar{\boldsymbol{\theta}}_{t,n}\right] \rangle, \tag{4.9}$$

indicating that the generalization error can be bounded by analyzing the bias and variance sub-problem. We now present the lemmas that bound the bias error.

**Lemma 11.** *The covariance* $\mathbb{E}\left[\bar{\boldsymbol{\theta}}^{bias}_{t,n} \otimes \bar{\boldsymbol{\theta}}^{bias}_{t,n}\right]$ *of the bias part of averaged iterate* $\bar{\boldsymbol{\theta}}^{bias}_{t,n}$ *satisfies:*

$$\mathbb{E}\left[\bar{\boldsymbol{\theta}}^{bias}_{t,n} \otimes \bar{\boldsymbol{\theta}}^{bias}_{t,n}\right] = \frac{1}{(n-t)^2}\Big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\Big) \cdot$$
$$(\mathcal{I} - \mathcal{B})^{-1}(\mathcal{B}^{t+1} - \mathcal{B}^{n+1})\left(\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0\right)$$
$$- \frac{1}{(n-t)^2}\sum_{j=t+1}^{n}\Big((\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}}^{n+1-j} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j}\Big)\mathcal{B}^{j}(\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0).$$

The quantity that needs to be bounded in the term above is $\mathcal{B}^{t+1}\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0$. Lemma 12 presents a result that can be applied recursively to bound $\mathcal{B}^{t+1}\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0$ ($= \mathcal{B}^{t+1}\boldsymbol{\theta}_0^{\text{bias}} \otimes \boldsymbol{\theta}_0^{\text{bias}}$ since $\boldsymbol{\theta}_0^{\text{bias}} = \boldsymbol{\theta}_0$).

**Lemma 12** (Bias contraction)**.** *For any two vectors* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, *let* $\boldsymbol{\theta} \overset{def}{=} \begin{bmatrix} \mathbf{x} - \mathbf{x}^* \\ \mathbf{y} - \mathbf{x}^* \end{bmatrix} \in \mathbb{R}^{2d}$. *We have:*

$$\langle \mathbf{G}, \mathcal{B}\left(\boldsymbol{\theta}\boldsymbol{\theta}^{\top}\right)\rangle \leq \left(1 - \frac{1}{9\sqrt{\kappa\widetilde{\kappa}}}\right)\langle \mathbf{G}, \boldsymbol{\theta}\boldsymbol{\theta}^{\top}\rangle$$

**Remarks:** (i) the matrices $\widetilde{\mathbf{G}}$ and $\widetilde{\mathbf{G}}^{\top}$ appearing in $\mathbf{G}$ are due to the fact that we prove contraction using the variables $\mathbf{x} - \mathbf{x}^*$ and $\mathbf{v} - \mathbf{x}^*$ instead of $\mathbf{x} - \mathbf{x}^*$ and $\mathbf{y} - \mathbf{x}^*$, as used in defining $\boldsymbol{\theta}$. (ii) The key novelty in Lemma 12 is that while standard analyses of accelerated gradient descent (in the exact first order oracle) use the potential function $\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{H}}^2 + \mu\|\mathbf{v} - \mathbf{x}^*\|_2^2$ (e.g. [123]), we consider it crucial for employing the potential function $\|\mathbf{x} - \mathbf{x}^*\|_2^2 + \mu\|\mathbf{v} - \mathbf{x}^*\|_{\mathbf{H}^{-1}}^2$ (this potential function is captured using the matrix $\mathbf{Z}$) to prove accelerated rates (of $\mathcal{O}\left(1/\sqrt{\kappa\widetilde{\kappa}}\right)$) for bias decay.

We now present the lemmas associated with bounding the variance error:

**Lemma 13.** *The covariance* $\mathbb{E}\left[\bar{\boldsymbol{\theta}}^{variance}_{t,n} \otimes \bar{\boldsymbol{\theta}}^{variance}_{t,n}\right]$ *of the variance error* $\bar{\boldsymbol{\theta}}^{variance}_{t,n}$ *satisfies:*

$$\begin{aligned}\mathbb{E}\left[\bar{\boldsymbol{\theta}}^{variance}_{t,n} \otimes \bar{\boldsymbol{\theta}}^{variance}_{t,n}\right] &= \frac{1}{n-t}\big(\mathcal{I} + (\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\big)(\mathcal{I}-\mathcal{B})^{-1}\widehat{\boldsymbol{\Sigma}} \\ &- \frac{1}{(n-t)^2}\big((\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-2}(\mathcal{A}_{\mathcal{L}} - \mathcal{A}_{\mathcal{L}}^{n+1-t}) + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-2}(\mathcal{A}_{\mathcal{R}}^{\top} - (\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-t})\big)(\mathcal{I}-\mathcal{B})^{-1}\widehat{\boldsymbol{\Sigma}} \\ &- \frac{1}{(n-t)^2}\big(\mathcal{I} + (\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\big)(\mathcal{I}-\mathcal{B})^{-2}(\mathcal{B}^{t+1} - \mathcal{B}^{n+1})\widehat{\boldsymbol{\Sigma}} \\ &+ \frac{1}{(n-t)^2}\sum_{j=t+1}^{n}\big((\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}}^{n+1-j} + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j}\big)(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{j}\widehat{\boldsymbol{\Sigma}}.\end{aligned}$$

The covariance of the stationary distribution of the iterates i.e., $\lim_{j\to\infty}\boldsymbol{\theta}_j^{\text{variance}}$ requires a precise bound to obtain statistically optimal error rates. Lemma 14 presents a bound on this quantity.

**Lemma 14** (Stationary covariance)**.** *The covariance of limiting distribution of* $\boldsymbol{\theta}^{variance}$ *satisfies:*

$$\mathbb{E}\left[\boldsymbol{\theta}^{variance}_{\infty} \otimes \boldsymbol{\theta}^{variance}_{\infty}\right] = (\mathbf{I}-\mathcal{B})^{-1}\widehat{\boldsymbol{\Sigma}} \preceq 5\sigma^2\Bigg((2/3)\cdot\big(\frac{1}{\tilde{\kappa}}\mathbf{H}^{-1}\big) + (5/6)\cdot(\delta\mathbf{I})\Bigg) \otimes \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}.$$

A crucial implication of this lemma is that the limiting final iterate $\boldsymbol{\theta}_\infty^{\text{variance}}$ has an excess risk $\mathcal{O}(\sigma^2)$. This result naturally lends itself to the (tail-)averaged iterate achieving the minimax optimal rate of $\mathcal{O}(d\sigma^2/n)$. Refer to the appendix B.5 and Lemma 41 for more details in this regard.

## *4.5 Conclusion*

This chapter introduces an accelerated stochastic gradient method, which presents the first improvement in achieving minimax rates faster than averaged SGD [99, 103, 94, 55]/Streaming SVRG [35] for the stochastic approximation problem of least squares regression. To obtain this result, the chapter presented the need to rethink what acceleration has to offer when working with a stochastic gradient oracle: these thought experiments indicated a need to consider a quantity that captured more fine grained problem characteristics. The statistical condition number (an affine invariant distributional quantity) is shown to characterize the improvements that acceleration offers in the stochastic first order oracle model.

In essence, this chapter presents the first known provable analysis of the claim that fast gradient methods are stable when dealing with statistical errors, in contrast to negative results in statistical and non-statistical settings [91, 95, 93, 42, 102, 113, 22, 27, 28, 125]. We hope that this chapter provides insights towards developing simple and effective accelerated stochastic gradient schemes for general convex and non-convex optimization problems.

# Chapter 5

# UNDERSTANDING THE BEHAVIOR OF MOMENTUM WITH STOCHASTIC GRADIENTS

## 5.1 Chapter Notes

This chapter presents joint work with Prateek Jain, Praneeth Netrapalli and Sham M. Kakade and is published in the International Conference on Learning Representations (ICLR), 2018. This chapter's presentation is a (very minor) variant of the published version of this paper. The contributions of this chapter are summarized as follows:

- This chapter presents an understanding of momentum based methods when used in conjunction with stochastic gradient methods. Momentum based methods were developed in [92] as a means to improve over gradient descent's convergence guarantees when encountering (deterministic i.e.) exact gradients of an objective. Despite their origin, momentum methods have shown immense performance improvements when working with mini-batch stochastic gradients for several Deep Learning problems of interest. This chapter delves into these questions and presents the (surprising) result that momentum based stochastic gradient descent methods do not offer any running time advantages (beyond constant factors) over standard SGD, especially when working with small constant sized mini-batches. This is in stark contrast to the behavior of Accelerated SGD, described in chapter 4, which is shown to provably improve over SGD irrespective of the batchsizes used.

- These improvements (or lack thereof) aren't restricted to certain problem classes. This chapter shows that, in the context of training several modern Neural Networks (including convolutional residual networks and fully connected networks) for various classes of problems, one does observe that (a) momentum based SGD methods do not offer much improvements with standard batchsizes typically used in practice, and, (b) accelerated SGD does present a noticeable improvement over SGD as well as momentum based SGD methods.

## 5.2 Introduction

First order optimization methods, which access a function to be optimized through its gradient or an unbiased approximation of its gradient, are the workhorses for modern large scale optimization problems, which include training the current state-of-the-art deep neural networks. Gradient descent [18] is the simplest first order method that is used heavily in practice. However, it is known that for the class of smooth convex functions as well as some simple non-smooth problems [84]), gradient descent is suboptimal [86] and there exists a class of algorithms called fast gradient/momentum based methods which achieve optimal convergence guarantees. The heavy ball method [92] and Nesterov's accelerated gradient descent [85] are two of the most popular methods in this category.

On the other hand, training deep neural networks on large scale datasets have been possible through the use of Stochastic Gradient Descent (SGD) [99], which samples a random subset of training data to compute gradient estimates that are then used to optimize the objective function. The advantages of SGD for large scale optimization and the related issues of tradeoffs between computational and statistical efficiency was highlighted in [12].

The above mentioned theoretical advantages of fast gradient methods [92, 85] (albeit for smooth convex problems) coupled with cheap to compute stochastic gradient estimates led to the influential work of [116], which demonstrated the empirical advantages possessed by SGD when augmented with the momentum machinery. This work has led to widespread adoption of momentum methods for training deep neural nets; so much so that, in the context of neural network training, gradient descent often refers to momentum methods.

But, there is a subtle difference between classical momentum methods and their implementation in practice – classical momentum methods work in the exact first order oracle model [86], i.e., they employ *exact gradients* (computed on the full training dataset), while in practice [116], they are implemented with *stochastic gradients* (estimated from a randomly sampled mini-batch of training data). This leads to a natural question:

*"Are momentum methods optimal even in the* **stochastic first order oracle** *(SFO) model, where we access stochastic gradients computed on a small constant sized minibatches (or a batchsize of* 1*?)"*

Even disregarding the question of optimality of momentum methods in the SFO model, it is not even known if momentum methods (say, [92, 85]) provide *any provable* improvement over SGD in this model. While these are open questions, a recent effort of [54] showed that improving upon SGD (in the stochastic first order oracle) is rather subtle as there exists problem instances in SFO model where it is not possible to improve upon SGD, even information theoretically. [54] studied a variant of Nesterov's accelerated gradient updates [87] for stochastic linear regression and show that their method improves upon SGD wherever it is information theoretically admissible. Through out this chapter, we refer to the algorithm of [54] as Accelerated Stochastic Gradient Method (ASGD) while we refer to a stochastic version of the most widespread form of Nesterov's method [85] as NAG; HB denotes a stochastic version of the heavy ball method [92]. Critically, while [54] shows that ASGD improves on SGD in any information-theoretically admissible regime, it is still not known whether HB and NAG can achieve a similar performance gain.

A key contribution of this chapter is to show that HB *does not* provide similar performance gains over SGD even when it is informationally-theoretically admissible. That is, we provide a problem instance where it is indeed possible to improve upon SGD (and ASGD achieves this improvement), but HB *cannot* achieve any improvement over SGD. We validate this claim empirically as well. In fact, we provide empirical evidence to the claim that NAG also do not achieve any improvement over SGD for several problems where ASGD can still achieve better rates of convergence.

This raises a question about why HB and NAG provide better performance than SGD in practice [116], especially for training deep networks. Our conclusion (that is well supported by our theoretical result) is that HB and NAG's improved performance is attributed to mini-batching and hence, these methods will often struggle to improve over SGD with small constant batch sizes. This is in stark contrast to methods like ASGD, which is designed to improve over SGD across both small or large mini-batch sizes. In fact, based on our experiments, we observe that on the task of training deep residual networks [47] on the cifar-10 dataset, we note that ASGD offers noticeable improvements by achieving $5-7\%$ better test error over HB and NAG even with commonly used batch sizes like 128 during the initial stages of the optimization.

### *5.2.1 Contributions*

The contributions of this chapter are as follows.

1. In Section 5.4, we prove that HB is *not optimal* in the SFO model. In particular, there exist linear regression problems for which the performance of HB (with *any* step size and momentum) is either the same or worse than that of SGD while ASGD improves upon both of them.

2. Experiments on several linear regression problems suggest that the suboptimality of HB in the SFO model is not restricted to special cases – it is rather widespread. Empirically, the same holds true for NAG as well (Section 5.6).

3. The above observations suggest that the only reason for the superiority of momentum methods in practice is mini-batching, which reduces the variance in stochastic gradients and moves the SFO closer to the exact first order oracle. This conclusion is supported by empirical evidence through training deep residual networks on cifar-10, with a batch size of 8 (see Section 5.6.3).

4. We present an intuitive and easier to tune version of ASGD (see Section 5.5) and show that ASGD can provide significantly faster convergence to a reasonable accuracy than

SGD, HB, NAG, while still providing favorable or comparable asymptotic accuracy as these methods, particularly on several deep learning problems.

Hence, the take-home message of this chapter is: *HB and NAG are not optimal in the SFO model. The only reason for the superiority of momentum methods in practice is mini-batching. ASGD provides a distinct advantage in training deep networks over SGD, HB and NAG.*

## 5.3 Notation

We denote matrices by bold-face capital letters and vectors by lower-case letters. $f(w) = 1/n \sum_i f_i(w)$ denotes the function to optimize w.r.t. model parameters $w$. $\nabla f(w)$ denotes exact gradient of $f$ at $w$ while $\widehat{\nabla} f_t(w)$ denotes a stochastic gradient of $f$. That is, $\widehat{\nabla} f_t(w_t) = \nabla f_{i_t}(w)$ where $i_t$ is sampled uniformly at random from $[1, \ldots, n]$. For linear regression, $f_i(w) = 0.5 \cdot (b_i - \langle w, a_i \rangle)^2$ where $b_i \in \Re$ is the target and $a_i \in \Re^d$ is the covariate, and $\widehat{\nabla} f_t(w_t) = -(b_t - \langle w_t, a_t \rangle) a_t$. In this case, $\mathbf{H} = \mathbb{E}\left[aa^\top\right]$ denotes the Hessian of $f$ and $\kappa = \frac{\lambda_1(\mathbf{H})}{\lambda_d(\mathbf{H})}$ denotes it's condition number.

Algorithm 4 provides a pseudo-code of HB method [92]. $w_t - w_{t-1}$ is the momentum term and $\alpha$ denotes the momentum parameter. Next iterate $w_{t+1}$ is obtained by a linear combination of the SGD update and the momentum term. Algorithm 5 provides pseudo-code of a stochastic version of the most commonly used form of Nesterov's accelerated gradient descent [85].

## 5.4 Suboptimality of Heavy Ball Method

In this section, we show that there exists linear regression problems where the performance of HB (Algorithm 4) is no better than that of SGD, while ASGD significantly improves upon SGD's performance. Let us now describe the problem instance.

**Algorithm 4** HB: Heavy ball with a SFO

```
Require: Initial $w_0$, stepsize $\delta$, momentum $\alpha$
1: $w_{-1} \leftarrow w_0$; $t \leftarrow 0$        /*Set $w_{-1}$ to $w_0$*/
2: while $w_t$ not converged do
3:     $w_{t+1} \leftarrow w_t - \delta \cdot \widehat{\nabla} f_t(w_t) + \alpha \cdot (w_t - w_{t-1})$
       /*Sum of stochastic gradient step and momentum*/
4:     $t \leftarrow t + 1$
5: end while
Ensure: $w_t$        /*Return the last iterate*/
```

**Algorithm 5** NAG: Nesterov's AGD with a SFO

```
Require: Initial $w_0$, stepsize $\delta$, momentum $\alpha$
1: $v_0 \leftarrow w_0$; $t \leftarrow 0$        /*Set $v_0$ to $w_0$*/
2: while $w_t$ not converged do
3:     $v_{t+1} \leftarrow w_t - \delta \cdot \widehat{\nabla} f_t(w_t)$ /*SGD step*/
4:     $w_{t+1} = (1 + \alpha)v_{t+1} - \alpha v_t$/*Sum of SGD step and previous iterate*/
5:     $t \leftarrow t + 1$
6: end while
Ensure: $w_t$        /*Return the last iterate*/
```

Fix $w^* \in \mathbb{R}^2$ and let $(a, b) \sim \mathcal{D}$ be a sample from the distribution such that:

$$a = \begin{cases} \sigma_1 \cdot z \cdot e_1 \text{ w.p. } 0.5 \\ \sigma_2 \cdot z \cdot e_2 \text{ w.p. } 0.5, \end{cases} \qquad \text{and} \qquad b = \langle w^*, a \rangle,$$

where $e_1, e_2 \in \mathbb{R}^2$ are canonical basis vectors, $\sigma_1 > \sigma_2 > 0$. Let $z$ be a random variable such that $\mathbb{E}[z^2] = 2$ and $\mathbb{E}[z^4] = 2c \geq 4$. Hence, we have: $\mathbb{E}\left[(a^{(i)})^2\right] = \sigma_i^2, \mathbb{E}\left[(a^{(i)})^4\right] = c\sigma_i^4$, for $i = 1, 2$. Now, our goal is to minimize:

$$f(w) \stackrel{\text{def}}{=} 0.5 \cdot \mathbb{E}\left[(\langle w^*, a \rangle - b)^2\right], \text{ Hessian } \mathbf{H} \stackrel{\text{def}}{=} \mathbb{E}\left[aa^\top\right] = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}.$$

Let $\kappa$ and $\tilde{\kappa}$ denote the *computational* and *statistical* condition numbers – see [54] for definitions. For the problem above, we have $\kappa = \frac{c\sigma_1^2}{\sigma_2^2}$ and $\tilde{\kappa} = c$. Then we obtain following convergence rates for SGD and ASGD when applied to the above given problem instance:

**Corollary 15** (of Theorem 1 of [55])**.** *Let $w_t^{SGD}$ be the $t^{th}$ iterate of SGD on the above problem with starting point $w_0$ and stepsize $\frac{1}{c\sigma_1^2}$. The error of $w_t^{SGD}$ can be bounded as,*

$$\mathbb{E}\left[f\left(w_t^{SGD}\right)\right] - f\left(w_*\right) \leq \exp\left(\frac{-t}{\kappa}\right)\left(f\left(w_0\right) - f\left(w_*\right)\right).$$

On the other hand, ASGD achieves the following superior rate.

**Corollary 16** (of Theorem 1 of [54])**.** *Let $w_t^{ASGD}$ be the $t^{th}$ iterate of ASGD on the above problem with starting point $w_0$ and appropriate parameters. The error of $w_t^{ASGD}$ can be bounded as,*

$$\mathbb{E}\left[f\left(w_t^{ASGD}\right)\right] - f\left(w_*\right) \leq \mathrm{poly}(\kappa)\exp\left(\frac{-t}{\sqrt{\kappa\tilde{\kappa}}}\right)\left(f\left(w_0\right) - f\left(w_*\right)\right).$$

Note that for a given problem/input distribution $\tilde{\kappa} = c$ is a constant while $\kappa = \frac{c\sigma_1^2}{\sigma_2^2}$ can be arbitrarily large. Note that $\kappa > \tilde{\kappa} = c$. Hence, ASGD improves upon rate of SGD by a factor of $\sqrt{\kappa}$. The following proposition, which is the main result of this section, establishes that HB (Algorithm 4) cannot provide a similar improvement over SGD as what ASGD offers. In fact, we show no matter the choice of parameters of HB, its performance does not improve over SGD by more than a constant.

**Proposition 17.** *Let $w_t^{HB}$ be the $t^{th}$ iterate of HB (Algorithm 4) on the above problem with starting point $w_0$. For* any *choice of stepsize $\delta$ and momentum $\alpha \in [0,1]$, $\exists T$ large enough such that $\forall t \geq T$, we have,*

$$\mathbb{E}\left[f\left(w_t^{HB}\right)\right] - f\left(w_*\right) \geq C(\kappa,\delta,\alpha)\cdot\exp\left(\frac{-500t}{\kappa}\right)\left(f\left(w_0\right) - f\left(w_*\right)\right),$$

*where $C(\kappa,\delta,\alpha)$ depends on $\kappa,\delta$ and $\alpha$ (but not on $t$).*

Thus, to obtain $\widehat{w}$ s.t. $\|\widehat{w} - w^*\| \leq \epsilon$, HB requires $\Omega(\kappa\log\frac{1}{\epsilon})$ samples and iterations. On the other hand, ASGD can obtain $\epsilon$-approximation to $w^*$ in $\mathcal{O}(\sqrt{\kappa}\log\kappa\log\frac{1}{\epsilon})$ iterations. We note that the gains offered by ASGD are meaningful when $\kappa > \mathcal{O}(c)$ [54]; otherwise, all the algorithms including SGD achieve nearly the same rates (upto constant factors). While

**Algorithm 6** Accelerated stochastic gradient descent – ASGD

```
Input: Initial $w_0$, short step $\delta$, long step parameter $\kappa \geq 1$, statistical advantage parameter $\xi \leq \sqrt{\kappa}$
1: $\bar{w}_0 \leftarrow w_0$; $t \leftarrow 0$                                   /*Set running average to $w_0$*/
2: $\alpha \leftarrow 1 - \frac{0.7^2 \cdot \xi}{\kappa}$                                   /*Set momentum value*/
3: while $w_t$ not converged do
4:     $\bar{w}_{t+1} \leftarrow \alpha \cdot \bar{w}_t + (1-\alpha) \cdot \Big(w_t - \frac{\kappa \cdot \delta}{0.7} \cdot \widehat{\nabla} f_t(w_t)\Big)$     /*Update the running average as a weighted average of previous running average and a long step gradient */
5:     $w_{t+1} \leftarrow \frac{0.7}{0.7+(1-\alpha)} \cdot \Big(w_t - \delta \cdot \widehat{\nabla} f_t(w_t)\Big) + \frac{1-\alpha}{0.7+(1-\alpha)} \cdot \bar{w}_{t+1}$     /*Update the iterate as weighted average of current running average and short step gradient*/
6:     $t \leftarrow t+1$
7: end while
Output: $w_t$                                   /*Return the last iterate*/
```

we do not prove it theoretically, we observe empirically that for the same problem instance, NAG also obtains nearly same rate as HB and SGD. We conjecture that a lower bound for NAG can be established using a similar proof technique as that of HB (i.e. Proposition 17). We conjecture that the constant in the lower bound described in Proposition 17 can be improved to some small number ($\leq 5$).

## *5.5 Algorithm*

We will now present and explain an intuitive version of ASGD (pseudo code in Algorithm 6). The algorithm takes three inputs: short step $\delta$, long step parameter $\kappa$ and statistical advantage parameter $\xi$. The short step $\delta$ is precisely the same as the step size in SGD, HB or NAG. For convex problems, this scales inversely with the smoothness of the function. The long step parameter $\kappa$ is intended to give an estimate of the ratio of the largest and smallest curvatures of the function; for convex functions, this is just the condition number. The statistical advantage parameter $\xi$ captures trade off between statistical and computational condition numbers – in the deterministic case, $\xi = \sqrt{\kappa}$ and ASGD is equivalent to NAG, while in the high stochasticity regime, $\xi$ is much smaller. The algorithm maintains two iterates: descent iterate $w_t$ and a running average $\bar{w}_t$. The running average is a weighted average of the previous average and a long gradient step from the descent iterate, while the descent iterate is updated as a convex combination of short gradient step from the descent iterate and the running average. The idea is that since the algorithm takes a long step as well as short step and an appropriate average of both of them, it can make progress on different directions at a similar pace. Appendix C.2 shows the equivalence between Algorithm 6 and ASGD as proposed in [54]. Note that the constant 0.7 appearing in Algorithm 6 has no special significance. [54] require it to be smaller than $\sqrt{1/6}$ but any constant smaller than 1 seems to work in practice.

## *5.6 Experiments*

We now present our experimental results exploring performance of SGD, HB, NAG and ASGD. Our experiments are geared towards answering the following questions:

- Even for linear regression, is the suboptimality of HB restricted to specific distributions in Section 5.4 or does it hold for more general distributions as well? Is the same true of NAG?

- What is the reason for the superiority of HB and NAG in practice? Is it because momentum methods have better performance that SGD for stochastic gradients or due to mini-batching? Does this superiority hold even for small minibatches?

- How does the performance of ASGD compare to that of SGD, HB and NAG, when training deep networks?

Section 5.6.1 and parts of Section 5.6.2 address the first two questions. Section 5.6.2 and 5.6.3 address Question 2 partially and the last question. We use Matlab to conduct experiments presented in Section 5.6.1 and use PyTorch for our deep networks related experiments. Pytorch code implementing the ASGD algorithm can be found at here [1].

### *5.6.1 Linear Regression*

In this section, we will present results on performance of the four optimization methods (SGD, HB, NAG, and ASGD) for linear regression problems. We consider two different class of linear regression problems, both in two dimensions. For a given condition number $\kappa$, we consider the following two distributions:

**Discrete**: $a = e_1$ w.p. 0.5 and $a = \frac{2}{\kappa} \cdot e_2$ with 0.5; $e_i$ is the $i^{th}$ standard basis vector.

**Gaussian** : $a \in \mathbb{R}^2$ is sampled from a Gaussian with covariance matrix $\begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\kappa} \end{bmatrix}$.

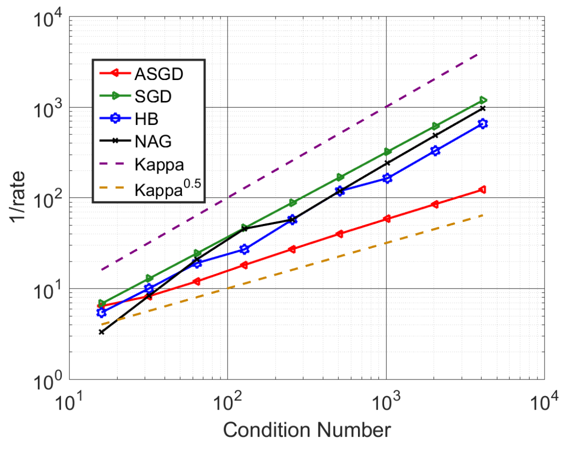
We fix a randomly generated $w^* \in \mathbb{R}^2$ and for both the distributions above, we let $b = \langle w^*, a \rangle$. We vary $\kappa$ from $\{2^4, 2^5, ..., 2^{12}\}$ and for each $\kappa$ in this set, we run 100 independent runs of all four methods, each for a total of $t = 5\kappa$ iterations. We define that the algorithm converges if there is no error in the second half (i.e. after $2.5\kappa$ updates) that exceeds the starting error - this is reasonable since we expect *geometric convergence* of the initial error.

Unlike ASGD and SGD, we do not know optimal learning rate and momentum parameters for NAG and HB in the stochastic gradient model. So, we perform a *grid search* over the values of the learning rate and momentum parameters. In particular, we lay a $10 \times 10$ grid in $[0, 1] \times [0, 1]$ for learning rate and momentum and run NAG and HB. Then, for each grid point, we consider the subset of 100 trials that converged and computed the final error using these. Finally, the parameters that yield the minimal error are chosen for NAG and HB, and these numbers are reported. We measure convergence performance of a method using:

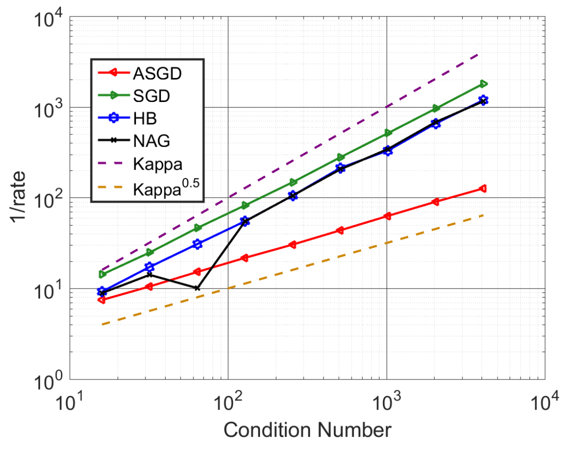
$$\text{rate} = \frac{\log(f(w_0)) - \log(f(w_t))}{t}, \tag{5.1}$$

We compute the rate (5.1) for all the algorithms with varying condition number $\kappa$. Given a rate vs $\kappa$ plot for a method, we compute it's *slope* (denoted as $\gamma$) using linear regression.

[1] `https://github.com/rahulkidambi/AccSGD`



(a) Discrete distribution



(b) Gaussian distribution

Figure 5.1: Plot of 1/rate (refer equation (5.1)) vs condition number ($\kappa$) for various methods for the linear regression problem.

Table 5.1 presents the estimated slopes (i.e. $\gamma$) for various methods for both the discrete and the Gaussian case. The slope values clearly show that the rate of SGD, HB and NAG have a nearly linear dependence on $\kappa$ while that of ASGD seems to scale linearly with $\sqrt{\kappa}$.

### *5.6.2 Deep Autoencoders for MNIST*

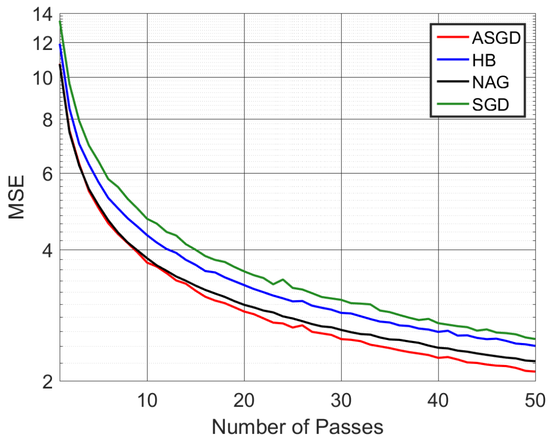
In this section, we present experimental results on training deep autoencoders for the mnist dataset, and we follow the setup of [49]. This problem is a standard benchmark for evaluating optimization algorithms e.g., [77, 116, 78, 98]. The network architecture follows previous work [49] and is represented as $784-1000-500-250-30-250-500-1000-784$ with the first and last 784 nodes representing the input and output respectively. All hidden/output nodes employ sigmoid activations except for the layer with 30 nodes which employs linear activations and we use MSE loss. We use the initialization scheme of [77], also employed in [116, 78]. We perform training with two minibatch sizes $-1$ and 8. The runs with minibatch size of 1 were run for 30 epochs while the runs with minibatch size of 8 were run

| Algorithm | Slope – discrete | Slope – Gaussian |
|---|---|---|
| SGD | 0.9302 | 0.8745 |
| HB | 0.8522 | 0.8769 |
| NAG | 0.98 | 0.9494 |
| ASGD | 0.5480 | 0.5127 |

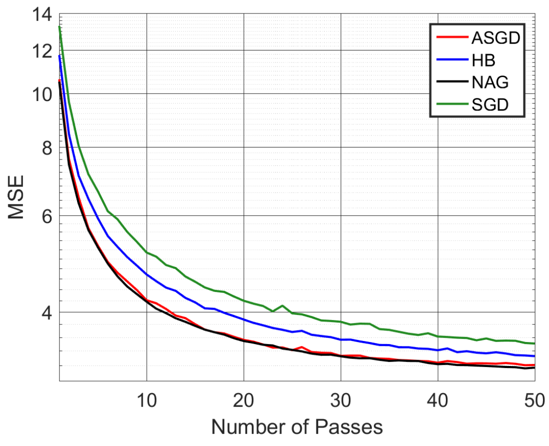
Table 5.1: Slopes (i.e. $\gamma$) obtained by fitting a line to the curves in Figure 5.1. A value of $\gamma$ indicates that the error decays at a rate of $\exp\left(\frac{-t}{\kappa^{\gamma}}\right)$. A smaller value of $\gamma$ indicates a faster rate of error decay.

for 50 epochs. For each of SGD, HB, NAG and ASGD, a grid search over learning rate, momentum and long step parameter (whichever is applicable) was done and best parameters were chosen based on achieving the smallest training error in the same protocol followed by [116]. The grid was extended whenever the best parameter fell at the edge of a grid. For the parameters chosen by grid search, we perform 10 runs with different seeds and averaged the results. The results are presented in Figures 5.2 and 5.3. Note that the final loss values reported are suboptimal compared to the published literature e.g., [116]; while [116] report results after 750000 updates with a large batch size of 200 (which implies a total of $750000 \times 200 = 150$M gradient evaluations), whereas, ours are after 1.8M updates of SGD with a batch size 1 (which is just 1.8M gradient evaluations). **Effect of minibatch sizes**: While HB and NAG decay the loss faster compared to SGD for a minibatch size of 8 (Figure 5.2), this superior decay rate does not hold for a minibatch size of 1 (Figure 5.3). This supports our intuitions from the stochastic linear regression setting, where we demonstrate that HB and NAG are suboptimal in the stochastic first order oracle model.

**Comparison of ASGD with momentum methods**: While ASGD performs slightly better than NAG for batch size 8 in the training error (Figure 5.2), ASGD decays the error at a faster rate compared to all the three other methods for a batch size of 1 (Figure 5.3).

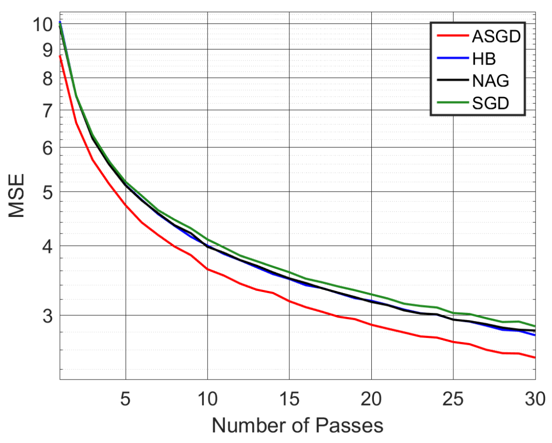


(a) Train Mean-Squared Error

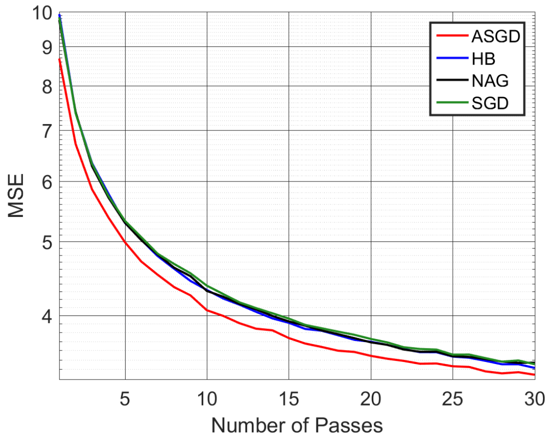


(b) Test Mean-Squared Error

Figure 5.2: Training loss (left) and test loss (right) while training deep autoencoder for mnist with minibatch size 8. Clearly, ASGD matches performance of NAG and outperforms SGD on the test data. HB also outperforms SGD.



(a) Train Mean-Squared Error



(b) Test Mean-Squared Error

Figure 5.3: Training loss (left) and test loss (right) while training deep autoencoder for mnist with minibatch size 1. Interestingly, SGD, HB and NAG, all decrease the loss at a similar rate, while ASGD decays at a faster rate.

### *5.6.3 Deep Residual Networks for CIFAR-10*

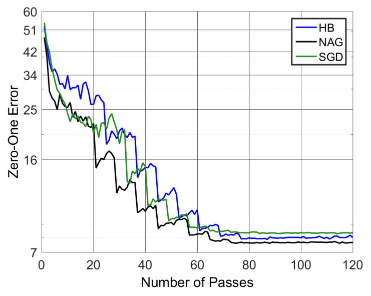
We will now present experimental results on training deep residual networks [46] with pre-activation blocks [47] for classifying images in cifar-10 [60]; the network we use has 44 layers (dubbed preresnet-44). The code for this section can be found at [2]. One of the most distinct characteristics of this experiment compared to our previous experiments is learning rate decay. We use a validation set based decay scheme, wherein, after every 3 epochs, we decay the learning rate by a certain factor (which we grid search on) if the validation zero one error does not decrease by at least a certain amount (precise numbers are provided in the appendix since they vary across batch sizes). Due to space constraints, we present only a subset of training error plots. See Appendix C.3.3 for some more plots on training errors.

**Effect of minibatch sizes**: Our first experiment tries to understand how the performance of HB and NAG compare with that of SGD and how it varies with minibatch sizes. Figure 5.4 presents the test zero one error for minibatch sizes of 8 and 128. While training with batch size 8 was done for 40 epochs, with batch size 128, it was done for 120 epochs. We perform a grid search over all parameters for each of these algorithms. See Appendix C.3.3 for details on the grid search parameters. We observe that final error achieved by SGD, HB and NAG are all very close for both batch sizes. While NAG exhibits a superior rate of convergence compared to SGD and HB for batch size 128, this superior rate of convergence disappears for a batch size of 8.

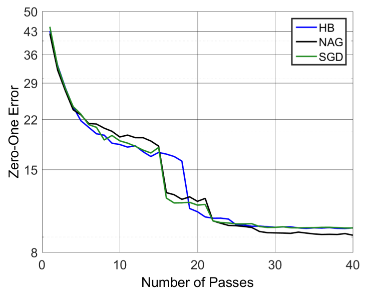
**Comparison of ASGD with momentum methods**: The next experiment tries to understand how ASGD compares with HB and NAG. The errors achieved by various methods when we do grid search over all parameters are presented in Table 5.2. Note that the final test errors for batch size 128 are better than those for batch size 8 since the former was run for 120 epochs while the latter was run only for 40 epochs (due to time constraints).

While the final error achieved by ASGD is similar/favorable compared to all other methods, we are also interested in understanding whether ASGD has a superior convergence

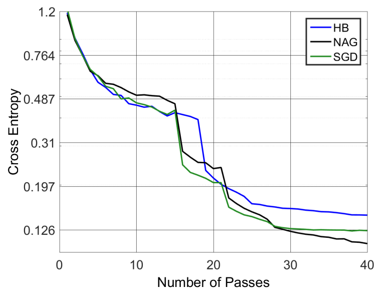
[2]`https://github.com/D-X-Y/ResNeXt-DenseNet`



(a) Test 0/1 error



(b) Test 0/1 error



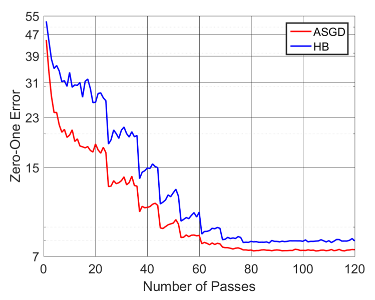
(c) Training Function Value

Figure 5.4: Test zero one loss for batch size 128 (left), batch size 8 (center) and training function value for batch size 8 (right) for SGD, HB and NAG.

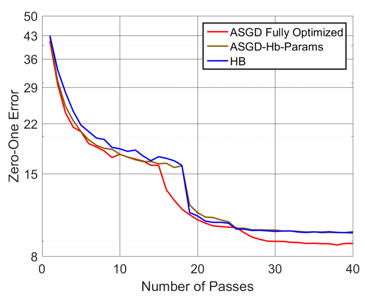
| Algorithm | Final test error – batch size 128 | Final test error – batch size 8 |
|---|---|---|
| SGD | $8.32 \pm 0.21$ | $9.57 \pm 0.18$ |
| HB | $7.98 \pm 0.19$ | $9.28 \pm 0.25$ |
| NAG | $7.63 \pm 0.18$ | $9.07 \pm 0.18$ |
| ASGD | $\mathbf{7.23 \pm 0.22}$ | $\mathbf{8.52 \pm 0.16}$ |

Table 5.2: Final test errors achieved by various methods for batch sizes of 128 and 8. The hyperparameters have been chosen by grid search.

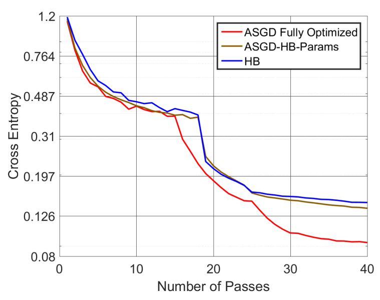
speed. For this experiment, we need to address the issue of differing learning rates used by various algorithms and different iterations where they decay learning rates. So, for each of HB and NAG, we choose the learning rate and decay factors by grid search, use these values for ASGD and do grid search only over long step parameter $\kappa$ and momentum $\alpha$ for ASGD. The results are presented in Figures 5.5 and 5.6. For batch size 128, ASGD decays error at a faster rate compared to both HB and NAG. For batch size 8, while we see a superior convergence of ASGD compared to NAG, we do not see this superiority over HB. The reason for this turns out to be that the learning rate for HB, which we also use for ASGD, turns out



(a) Test $0-1$ Error



(b) Test $0-1$ Error



(c) Training Function Value

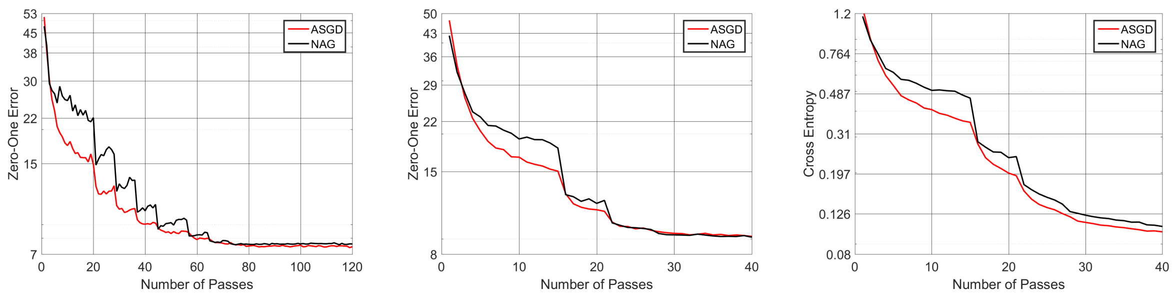
Figure 5.5: Test zero one loss for batch size 128 (left), batch size 8 (center) and training function value for batch size 8 (right) for ASGD compared to HB. In the above plots, both ASGD and ASGD-Hb-Params refer to ASGD run with the learning rate and decay schedule of HB. ASGD-Fully-Optimized refers to ASGD where learning rate and decay schedule were also selected by grid search.

to be quite suboptimal for ASGD. So, for batch size 8, we also compare fully optimized (i.e., grid search over learning rate as well) ASGD with HB. The superiority of ASGD over HB is clear from this comparison. These results suggest that ASGD decays error at a faster rate compared to HB and NAG across different batch sizes.

## 5.7 *Related Work*

**First order oracle methods**: The primary method in this family is Gradient Descent (GD) [18]. As mentioned previously, GD is suboptimal for smooth convex optimization [86], and this is addressed using momentum methods such as the Heavy Ball method [92] (for quadratics), and Nesterov's Accelerated gradient descent [85].

**Stochastic first order methods and noise stability**: The simplest method employing the SFO is SGD [99]; the effectiveness of SGD has been immense, and its applicability goes well beyond optimizing convex objectives. Accelerating SGD is a tricky proposition given the instability of fast gradient methods in dealing with noise, as evidenced by several



(a) Test $0-1$ Error (b) Test $0-1$ Error (c) Training Function Value

Figure 5.6: Test zero one loss for batch size 128 (left), batch size 8 (center) and training function value for batch size 8 (right) for ASGD compared to NAG. In the above plots, ASGD was run with the learning rate and decay schedule of NAG. Other parameters were selected by grid search.

negative results which consider statistical [95, 93, 102], numerical [91, 42] and adversarial errors [22, 28]. A result of [54] developed the first provably accelerated SGD method for linear regression which achieved minimax rates, inspired by a method of [87]. Schemes of [38, 39, 30], which indicate acceleration is possible with noisy gradients do not hold in the SFO model satisfied by algorithms that are run in practice (see [54] for more details).

While HB [92] and NAG [85] are known to be effective in case of exact first order oracle, for the SFO, the theoretical performance of HB and NAG is not well understood.

**Understanding Stochastic Heavy Ball:** Understanding HB's performance with inexact gradients has been considered in efforts spanning several decades, in many communities like controls, optimization and signal processing. [93] considered HB with noisy gradients and concluded that the improvements offered by HB with inexact gradients vanish unless strong assumptions on the inexactness was considered; an instance of this is when the variance of inexactness decreased as the iterates approach the minimizer. [95, 102, 113] suggest that the improved non-asymptotic rates offered by stochastic HB arose at the cost of worse asymptotic behavior. We resolve these unquantified improvements on rates as being just constant factors over SGD, in stark contrast to the gains offered by ASGD. [74] state their method as Stochastic HB but require stochastic gradients that nearly behave as exact gradients; indeed, their rates match that of the standard HB method [92], but this is possible only with very large batchsizes [54].

**Accelerated and Fast Methods for finite-sums:** There have been developments pertaining to faster methods for finite-sums (also known as offline stochastic optimization): amongst these are methods such as SDCA [108], SAG [101], SVRG [57], SAGA [24], which offer linear convergence rates for strongly convex finite-sums, improving over SGD's sub-linear rates [97]. These methods have been improved using accelerated variants [109, 34, 72, 23, 2]. Note that these methods require storing the entire training set in memory and taking multiple passes over the same for guaranteed progress. Furthermore, these methods require computing a batch gradient or require memory requirements (typically $\Omega(|\text{ training data points}|)$). For deep learning problems, data augmentation is often deemed necessary for achieving good performance; this implies computing quantities such as batch gradient (or storage necessities) over this augmented dataset is often infeasible. Such requirements are mitigated by the use of simple streaming methods such as SGD, ASGD, HB, NAG. For other technical distinctions between the offline and online stochastic methods refer to [35].

**Practical methods for training deep networks**: Momentum based methods employed with stochastic gradients [116] have become standard and popular in practice. These schemes tend to outperform standard SGD on several important practical problems. As previously mentioned, we attribute this improvement to effect of mini-batching rather than improvement offered by HB or NAG in the SFO model. Schemes such as Adagrad [32], RMSProp [119], Adam [59] represent an important and useful class of algorithms. The advantages offered by these methods are orthogonal to the advantages offered by fast gradient methods; it is an important direction to explore augmenting these methods with ASGD as opposed to standard HB or NAG based acceleration schemes.

[20] proposed Entropy-SGD, which is an altered objective that adds a local strong convexity term to the actual empirical risk objective, with an aim to improve generalization.

However, we do not understand convergence rates for convex problems or the generalization ability of this technique in a rigorous manner. [20] propose to use SGD in their procedure but mention that they employ the HB/NAG method in their implementation for achieving better performance. Naturally, we can use ASGD in this context. Path normalized SGD [89] is a variant of SGD that alters the metric on which the weights are optimized. As noted in their paper, path normalized SGD could be improved using HB/NAG (or even ASGD).

## *5.8 Conclusions and Future Directions*

In this chapter, we show that the performance gain of HB over SGD in stochastic setting is attributed to mini-batching rather than the algorithm's ability to *accelerate* with stochastic gradients. Concretely, we provide a formal proof that for several *easy* problem instances, HB does not outperform SGD despite large condition number of the problem; we observe this trend for NAG in our experiments. In contrast, ASGD [54] provides significant improvement over SGD for these problem instances. We observe similar trends when training a resnet on cifar-10 and an autoencoder on mnist. This chapter motivates several directions such as understanding the behavior of ASGD on domains such as NLP, and developing automatic momentum tuning schemes [126].

# Chapter 6

# SGD'S FINAL ITERATE – MINIMAX OPTIMALITY AND STEPSIZE SCHEMES

## *6.1 Chapter Notes*

This chapter presents joint work with Rong Ge, Praneeth Netrapalli and Sham M. Kakade and is currently a pre-print [37]. The contributions of this chapter are summarized as follows:

- This chapter shows that SGD's final iterate with a polynomially decaying stepsize scheme is highly sub-optimal compared to the minimax rate. This is shown for least squares regression objective (with/without strong convexity).

- This chapter also shows that a geometrically decaying stepsize routine obtains near optimal rates on the final iterate of SGD for least squares regression (with/without strong convexity), given the knowledge of the end time.

- This chapter also indicates that in the limiting (anytime sense), SGD *has* to query highly sub-optimal iterates infinitely often (i.e. in a $\limsup$ sense).

- The improvements of the stepdecay scheme over standard polynomially decaying stepsizes is shown to manifest even in practice when considering training of a near state of the art convolutional residual network on the cifar-10 dataset.

## *6.2 Introduction*

Large scale machine learning relies almost exclusively on stochastic optimization methods [12], which include stochastic gradient descent (SGD) [99] and its variants [32, 57]. In this chapter, we restrict our attention to the SGD algorithm where we are concerned with the behavior of the final iterate (i.e. the last point when we terminate the algorithm). A majority of (minimax optimal) theoretical results for SGD focus on polynomially decaying stepsizes [26, 97, 65], [16, chap. 6] (or constant stepsizes [7, 25, 55] for the case of least squares regression) coupled with iterate averaging [103, 94] to achieve minimax optimal rates of convergence. However, practical SGD implementations typically return the final iterate of a stochastic gradient procedure. This line of work in theory (based on iterate averaging) and its discrepancy with regards to practice leads to the question with regards to the behavior of SGD's final iterate. Indeed, this question has motivated several efforts in stochastic convex optimization literature as elaborated below.

**Non-Smooth Stochastic Optimization:** The work of [111] raised the question with regards to the behavior of SGD's final iterate for non-smooth stochastic optimization. The work of [112] answered this question, indicating that SGD's final iterate with polynomially decaying stepsizes achieves near minimax rates (up to log factors) in an anytime (i.e. in a limiting) sense (when number of iterations SGD is run for is not known in advance), both with/without strong convexity. Under specific choices of step size sequences, [112]'s result on SGD's final iterate is tight owing to the recent work of [43]. More recently [56] presented an approach indicating that a more nuanced stepsize sequence serves to achieve minimax rates (up to constant factors) for the non-smooth stochastic optimization setting when the end time $T$ is known in advance.

**Least Squares Regression (LSR):** In contrast to the non-smooth setting, the state of our understanding of SGD's final iterate for smooth stochastic convex optimization, or, say, the streaming least squares regression setting is far less mature – this gap motivates this chapter's contributions. In particular, this chapter studies SGD's final iterate behavior under various stepsize choices for least squares regression (with and without strong convexity). The use of SGD's final iterate for the least mean squares objective has featured in several efforts [121, 95, 102], [122, chap. $5-9$], but these results *do not* achieve minimax rates of convergence, which leads to the following question:

" Can polynomially decaying stepsizes (known to achieve minimax rates when coupled with iterate averaging [103, 94]) offer minimax optimal rates on SGD's *final* iterate when optimizing the streaming least squares regression objective? If not, is there *any* other family of stepsizes that can guarantee minimax rates on the final iterate of stochastic gradient descent? "

This chapter presents progress on answering the above question − refer to contributions below for more details. Note that the oracle model employed by this chapter (to quantify SGD's final iterate behavior) has featured in a string of recent results that present a non-asymptotic understanding of SGD for least squares regression, with the caveat being that these results crucially rely on *iterate averaging* with constant stepsize sequences [7, 25, 55, 54, 53, 88].

**Our contributions:** This chapter establishes upper and lower bounds on the behavior of SGD's final iterate, as run with both standard polynomially decaying stepsizes as well as *step decay* schedules which tends to cut the stepsize by a constant factor after every constant number of epochs (see Algorithm 7), by considering the problem of streaming least squares regression (with and without strong convexity). Our main result indicates that step decay schedules offer significant improvements in achieving near minimax rates over polynomially decaying stepsizes in the known horizon case (when the end time $T$ is known in advance). Figure 6.1 illustrates that this difference is evident (empirically) even when optimizing a two-dimensional convex quadratic. Table 6.1 provides a summary. Finally, we present results that indicate the subtle (yet significant) differences between the known time horizon case and the anytime (i.e. the limiting) behavior of SGD's final iterate (see below).

Our main contributions are as follows:

- *Sub-optimality of polynomially decaying learning rate schemes:* For the strongly convex least squares case, this chapter shows that the final iterate of a polynomially decaying stepsize scheme (i.e. with $\eta_t \propto 1/t^{\alpha}$, with $\alpha \in [0.5, 1]$) is off the statistical minimax rate $d\sigma^2/T$ by a factor of the *condition number* of the problem. For the non-strongly convex

**Algorithm 7** Step Decay scheme

```
Require: Initial vector $\mathbf{w}$, starting learning rate $\eta_0$, number of steps $T$
1: for $\ell = 1 \cdots \log T$ do
2:     $\eta_\ell \leftarrow \eta_0 / 2^\ell$
3:     for $t = 1 \cdots T/\log T$ do
4:         $\mathbf{w} \leftarrow \mathbf{w} - \eta_\ell \widehat{\nabla} f(\mathbf{w})$
5:     end for
6: end for
Ensure: $\mathbf{w}$
```

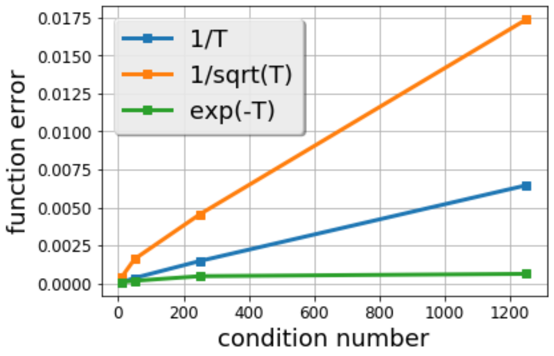


Figure 6.1: (Left) The Step Decay scheme for stochastic gradient descent. Note that the algorithm requires just two parameters - the starting learning rate $\eta_0$ and the number of iterations $T$.
(Right) Plot of function value error (in log scale) of the final iterate vs. condition number $\kappa$ for polynomially decaying i.e., equation(6.5), equation(6.6) and the smoothed geometrically decaying (i.e. exponentially decaying) step-sizes equation(6.7) for the 2-d quadratic problem (equation (6.1)), which also captures the behavior of a 2-d linear regression problem. The condition number $\kappa$ is varied as $\{10, 50, 250, 1250\}$. Exhaustive grid search is performed for all stepsize parameters ($\eta_0$ and $b$ in equation (6.5), (6.6), (6.7)). Initial error of the algorithm is $\mathcal{O}\left(d\sigma^2\right)$ and the algorithm is run for a total of $25 \times \kappa_{\max} = 25 \times 1250$ steps and averaged over 10 random seeds. Observe that the final iterate's error grows linearly as a function of the condition number $\kappa$ for the polynomially decaying stepsize schemes, whereas, the error grows only logarithmically in $\kappa$ for the smoothed geometric stepsize scheme. For details, refer to section D.5.1 in Appendix D.5.

case of least squares, we show that *any* polynomially decaying stepsize can achieve a rate no better than $d\sigma^2/\sqrt{T}$ (up to log factors), while the statistical minimax rate is $d\sigma^2/T$.

- *Near-optimality of the step-decay scheme:* Given a fixed end time $T$, the step-decay scheme (Algorithm 7) presents a final iterate that is off the statistical minimax rate by just a $\log(T)$ factor for optimizing both strongly convex and non-strongly convex case of least squares regression [1], thus indicating vast improvements over polynomially decaying stepsize schedules. We note here that our Theorem 19 for the non-strongly case offers a rate on the initial error (i.e., the bias term) that is off the best known rate [7] (that employs iterate averaging) by a dimension factor. That said, Algorithm 7 is rather straightforward and employs the knowledge of just an initial learning rate and number of iterations for its implementation.

- *SGD has to query bad iterates infinitely often:* For the case of optimizing strongly convex least squares regression, this chapter shows that any stochastic gradient procedure (in a $\limsup$ sense) *must* query sub-optimal iterates (off by nearly a condition number) infinitely often.

- Complementary to our theoretical results for the stochastic linear regression, we evaluate the empirical performance of different learning rate schemes when training a residual network on the cifar-10 dataset and observe that the continuous variant of step decay schemes (i.e. an exponential decay) indeed compares favorably to polynomially decaying stepsizes.

While the upper bounds established in this chapter (section 6.4.2) merit extensions towards broader smooth convex functions (with/without strong convexity), the lower bounds established in sections 6.4.1, 6.4.3 present implications towards broader classes of smooth

[1]This dependence can be improved to log of the condition number of the problem (for the strongly convex case) using a more refined stepsize decay scheme.

stochastic convex optimization. Even in terms of upper bounds, note that there are fewer results on non-asymptotic behavior of SGD (beyond least squares) when working in the oracle model considered in this chapter (see below). [5, 7, 6, 81] are exceptions, yet they do not achieve minimax rates on appropriate problem classes; [35] does not work in standard stochastic first order oracle model [83, 1], so their work is not directly comparable to examine extensions towards broader classes.

As a final note, this chapter's result on the sub-optimality of standard polynomially decaying stepsizes for classes of smooth and strongly convex optimization doesn't contradict the (minimax) optimality results in stochastic approximation [94]. Iterate averaging coupled with polynomially decaying learning rates (or constant learning rates for least squares [7, 25, 55]) does achieve minimax rates [103, 94]. However, as mentioned previously, this chapter deals with SGD's final iterate behavior (i.e. without iterate averaging), since this bears more relevance towards practice.

**Related work:** [99] introduced the stochastic approximation problem and Stochastic Gradient Descent (SGD). They present conditions on stepsize schemes satisfied by asymptotically convergent algorithms: these schemes are referred to as "convergent" stepsize sequences. [103, 94] proved the asymptotic optimality of iterate averaged SGD with larger stepsize sequences. In terms of oracle models and notions of optimality, there exists two lines of thought (see also [54]):

*Towards statistically optimal estimation procedures:* The goal of this line of thought is to match the excess risk of the statistically optimal estimator [4, 94], [69, chap. $5-6$] on every problem instance. Several efforts consider SGD in this oracle [5, 6, 29, 35, 81] presenting non-asymptotic results, often with iterate averaging. With regards to least squares, [7, 25, 35, 55, 54, 88] use constant step-size SGD with iterate averaging to achieve minimax rates (on a per-problem basis) in this oracle model. SGD's final iterate behavior for least squares has featured in several efforts in the signal processing/controls literature [121, 80, 95, 102, 113], [122, chap. $5-9$], without achieving minimax rates. This chapter works in this oracle model and analyzes SGD's final iterate behavior with various stepsize choices.

*Towards optimality under bounded noise assumptions:* The other line of thought presents algorithms with access to stochastic gradients satisfying bounded noise assumptions, aiming to match lower bounds provided in [83, 96, 1]. Asymptotic properties of "convergent" stepsize schemes have been studied in great detail [62, chap. 1,3,6], [8, chap. 1,3], [73, part 2], [64, chap. 5,10,11], [11, chap. 1,2,8] [9, 66]. Using iterate averaged SGD, efforts of [26, 65, 97, 38, 39, 45, 30], [16, chap. 6] achieve minimax rates for various problem classes non-asymptotically. [3] presents an alternative approach (compared to this chapter) towards minimizing the gradient norm with access to stochastic gradients. [45] use a doubling argument (and return a *random* iterate) to remove log factors in convergence rates - this is different from this chapter since it doubles epoch lengths every time it halves the learning rate (ours keeps the epoch lengths fixed and returns the final iterate). As mentioned before, [112] achieves anytime optimal rates (upto a $\log T$ factor) with the final iterate of an SGD procedure, and this is shown to be tight with the recent work of [43]. [56] achieve minimax rates on the final iterate using a nuanced stepsize scheme when the number of iterations is fixed in advance.

**Chapter organization:** Section 6.3 describes notation and problem setup. Section 6.4 presents our results on the sub-optimality of polynomial decay schemes and the near optimality of the step decay scheme. Section 6.4.3 presents results on the anytime behavior of SGD (i.e. the asymptotic/infinite horizon case). Section 6.5 presents experimental results and Section 6.6 presents conclusions.

## *6.3 Problem Setup*

**Notation**: We present the setup and associated notation in this section. We represent scalars with normal font $a, b, L$ etc., vectors with boldface lowercase characters $\mathbf{a}$ etc. and matrices with boldface uppercase characters $\mathbf{A}$ etc. We represent positive semidefinite (PSD) ordering between two matrices using $\succeq$. The symbol $\gtrsim$ represents that the inequality holds for some universal constant.

We consider here the minimization of the following expected square loss objective:

$$\min_{\mathbf{w}} f(\mathbf{w}) \text{ where } f(\mathbf{w}) \overset{\text{def}}{=} \tfrac{1}{2}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[(y - \mathbf{w}^\top\mathbf{x})^2]. \tag{6.1}$$

Note that the hessian of the objective $\mathbf{H} \overset{\text{def}}{=} \nabla^2 f(\mathbf{w}) = \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right]$. We are provided access to stochastic gradients obtained by sampling a new example $(\mathbf{x}_t, y_t) \sim \mathcal{D}$. These examples satisfy:

$$y = \langle \mathbf{w}^*, \mathbf{x}\rangle + \epsilon,$$

where, $\epsilon$ is the noise on the example pair $(\mathbf{x}, y) \sim \mathcal{D}$ and $\mathbf{w}^*$ is a minimizer of the objective $f(\mathbf{w})$. Given an initial iterate $\mathbf{w}_0$ and stepsize sequence $\{\eta_t\}$, our stochastic gradient update is:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \widehat{\nabla} f(\mathbf{w}_{t-1}); \qquad \widehat{\nabla} f(\mathbf{w}_t) = -(y_t - \langle \mathbf{w}_t, \mathbf{x}_t\rangle)\cdot \mathbf{x}_t. \tag{6.2}$$

We assume that the noise $\epsilon = y - \langle \mathbf{w}^*, \mathbf{x}\rangle \;\; \forall \;\; (\mathbf{x}, y) \sim \mathcal{D}$ satisfies the following condition:

$$\Sigma \overset{\text{def}}{=} \mathbb{E}\left[\widehat{\nabla} f(\mathbf{w}^*)\widehat{\nabla} f(\mathbf{w}^*)^\top\right] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[(y - \langle \mathbf{w}^*, \mathbf{x}\rangle)^2 \mathbf{x}\mathbf{x}^\top] \preceq \sigma^2 \mathbf{H}. \tag{6.3}$$

Next, assume that covariates $\mathbf{x}$ satisfy the following fourth moment inequality:

$$\mathbb{E}\left[\|\mathbf{x}\|^2 \mathbf{x}\mathbf{x}^\top\right] \preceq R^2\ \mathbf{H} \tag{6.4}$$

This assumption is satisfied, say, when the norm of the covariates $\sup \|\mathbf{x}\|^2 < R^2$, but is true more generally. Finally, note that both (6.3) and (6.4) are general and are used in recent works [7, 55] that present a sharp analysis of SGD for streaming least squares problem. Next, we denote by

$$\mu \overset{\text{def}}{=} \lambda_{\min}(\mathbf{H}), \quad L \overset{\text{def}}{=} \lambda_{\max}(\mathbf{H}), \text{ and }, \kappa \overset{\text{def}}{=} R^2/\mu$$

the smallest eigenvalue, largest eigenvalue and condition number of $\mathbf{H}$ respectively. $\mu > 0$ in the strongly convex case but not necessarily so in the non-strongly convex case (in section 6.4 and beyond, the non-strongly case is referred to as the "smooth" case). Let

$\mathbf{w}^* \in \operatorname{argmin}_{\mathbf{w}\in\mathbb{R}^d} f(\mathbf{w})$. The excess risk of an estimator $\mathbf{w}$ is $f(\mathbf{w}) - f(\mathbf{w}^*)$. Given $t$ accesses to the stochastic gradient oracle in equation (6.2), any algorithm that uses these stochastic gradients and outputs $\widehat{\mathbf{w}}_t$ has sub-optimality that is lower bounded by $\frac{\sigma^2 d}{t}$. More concretely, we have that [120, chap. 5,7,8]

$$\lim_{t\to\infty} \frac{\mathbb{E}\left[f(\widehat{\mathbf{w}}_t)\right] - f(\mathbf{w}^*)}{\sigma^2 d/t} \geq 1\,.$$

The rate of $(1 + o(1)) \cdot \sigma^2 d/t$ is achieved using iterate averaged SGD [103, 94] with constant stepsizes [7, 25, 55]. This rate of $\sigma^2 d/t$ is called the statistical minimax rate.

## 6.4 Main results

Sections 6.4.1, 6.4.2 consider the fixed time horizon setting; the former presents the significant sub-optimality of polynomially decaying stepsizes on SGD's final iterate behavior, the latter section presenting the near-optimality of SGD's final iterate. Section 6.4.3 presents negative results on SGD's final iterate behavior (irrespective of stepsizes employed), in the anytime (i.e. limiting) sense.

### *6.4.1 Suboptimality of polynomial decay schemes*

This section begins by showing that there exist problem instances where polynomially decaying stepsizes considered stochastic approximation theory [99, 94] i.e., those of the form $\frac{a}{b+t^\alpha}$, for any choice of $a, b > 0$ and $\alpha \in [0.5, 1]$ are significantly suboptimal (by a factor of the condition number of the problem, or by $\sqrt{T}$ in the smooth case) compared to the statistical minimax rate [62, chap. 1,3,6].

**Theorem 18.** *Under assumptions* (6.3), (6.4), *there exists a class of problem instances where the following lower bounds on excess risk hold on SGD's final iterate with polynomially decaying stepsizes when given access to the oracle as written in equation* (6.2).

***Strongly convex case:*** *Suppose $\mu > 0$. For any condition number $\kappa$, there exists a least squares problem instance with initial suboptimality $f(\mathbf{w}_0) - f(\mathbf{w}^*) \leq \sigma^2 d$ such that, for any*

*$T \geq \kappa^{\frac{4}{3}}$, and for all $a, b \geq 0$ and $0.5 \leq \alpha \leq 1$, and for the learning rate scheme $\eta_t = \frac{a}{b+t^\alpha}$, we have*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \geq \exp\left(-\frac{T}{\kappa \log T}\right)\left(f(\mathbf{w}_0) - f(\mathbf{w}^*)\right) + \frac{\sigma^2 d}{64} \cdot \frac{\kappa}{T}.$$

***Smooth case**: For any fixed $T > 1$, there exists a least squares problem instance such that, for all $a, b \geq 0$ and $0.5 \leq \alpha \leq 1$, and for the learning rate scheme $\eta_t = \frac{a}{b+t^\alpha}$, we have*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \geq \left(L \cdot \|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sigma^2 d\right) \cdot \frac{1}{\sqrt{T}\log T}.$$

For both cases (with/without strong convexity), the minimax rate is $\sigma^2 d/T$. In the strongly convex case, SGD's final iterate with polynomially decaying stepsizes pays a suboptimality factor of $\Omega(\kappa)$, whereas, in the smooth case, SGD's final iterate pays a suboptimality factor of $\Omega\left(\frac{\sqrt{T}}{\log T}\right)$.

### *6.4.2 Near optimality of Step Decay schemes*

Given the knowledge of an end time $T$ when the algorithm is terminated, this section presents the step decay schedule (Algorithm 7), which offers significant improvements over standard polynomially decaying stepsizes, and obtains near minimax rates (off by only a $\log(T)$ factor).

**Theorem 19.** *Suppose we are given access to the stochastic gradient oracle* (6.2) *satisfying Assumptions* (6.3) *and* (6.4)*. Running Algorithm 7 with an initial stepsize of $\eta_1 = 1/(2R^2)$ allows the algorithm to achieve the following excess risk guarantees.*

- ***Strongly convex case**: Suppose $\mu > 0$. We have:*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \leq 2 \cdot \exp\left(-\frac{T}{2\kappa \log T \log \kappa}\right)\left(f(\mathbf{w}_0) - f(\mathbf{w}^*)\right) + 4\sigma^2 d \cdot \frac{\log T}{T}.$$

- ***Smooth case**: We have:*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \leq 2 \cdot \left(R^2 d \cdot \|\mathbf{w}_0 - \mathbf{w}^*\|^2 + 2\sigma^2 d\right) \cdot \frac{\log T}{T}$$

While Theorem 19 presents significant improvements over polynomial decay schemes, as mentioned in the contributions, the above result presents a worse rate on the initial error (by a dimension factor) in the smooth case (i.e. non-strongly convex case), compared to the best known result [7], which relies heavily on iterate averaging to remove this factor. It is an open question with regards to whether this factor can actually be improved or not. Furthermore, comparing the initial error dependence between the lower bound for the smooth case (Theorem 18) with the upper bound for the step decay scheme, we believe that the dependence on the smoothness $L$ should be improved to one on the $R^2$.

In terms of the variance, however, note that the polynomial decay schemes are plagued by a polynomial dependence on the condition number $\kappa$ (for the strongly convex case), and are off the minimax rate by a $\sqrt{T}$ factor (for the smooth case). The step decay schedule, on the other hand, is off the minimax rate [103, 94], [120, chap. 5,7,8] by only a $\log(T)$ factor. It is worth noting that Algorithm 7 admits an efficient implementation in that it requires the knowledge only of $R^2$ (similar to iterate averaging results [7, 55]) and the end time $T$. Finally, note that this $\log T$ factor can be improved to a $\log \kappa$ factor for the strongly convex case by using an additional polynomial decay before switching to the Step Decay scheme.

**Proposition 20.** *Suppose we are given access to the stochastic gradient oracle* (6.2) *satisfying Assumptions* (6.3) *and* (6.4)*. Let $\mu > 0$ and let $\kappa \geq 2$. For any problem and fixed time horizon $T/\log T > 5\kappa$, there exists a learning rate scheme that achieves*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \leq 2\exp(-T/(6\kappa\log\kappa)) \cdot (f(\mathbf{w}_0) - f(\mathbf{w}^*)) + 100\log_2\kappa \cdot \frac{\sigma^2 d}{T}.$$

In order to have improved the dependence on the variance from $\log(T)$ (in Theorem 19) to $\log(\kappa)$ (in Proposition 20), we also require access to the strong convexity parameter $\mu$, similar to results known for general strongly convex case [97, 65, 112, 56], [16, chap. 6].

As a final remark, note that this section's results (on step decay schemes) assumed the knowledge of a fixed time horizon $T$. In contrast, most results SGD's averaged iterate obtain anytime (i.e., limiting/infinite horizon) guarantees. Can we hope to achieve such guarantees with the final iterate?

### *6.4.3 SGD queries bad points infinitely often*

This section shows that obtaining near statistical minimax rates with the *final* iterate is not possible when the time horizon $T$ is unknown. More concretely, we show that irrespective of the learning rate sequence employed, SGD *requires* to query a point with sub-optimality at least $\Omega(\kappa/\log\kappa)\cdot\sigma^2 d/T$ for infinitely many time steps $T$.

**Theorem 21.** *Suppose we are given access to a stochastic gradient oracle* (6.2) *satisfying Assumption* (6.3) *and* (6.4)*. There exists a universal constant $C>0$, and a problem instance, such that for SGD algorithm with any $\eta_t \le 1/2R^2$ for all $t$*[2]*, we have*

$$\limsup_{T\to\infty} \frac{\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*)}{(\sigma^2 d/T)} \ge C\frac{\kappa}{\log(\kappa+1)}.$$

The bad points guaranteed to exist by Theorem 21 are not rare. One can in fact show that such points occur at least once in $\mathcal{O}\left(\frac{\kappa}{\log\kappa}\right)$ iterations. Refer to [37] for more details.

## *6.5 Experimental Results*

We present experimental validation on the suitability of the Step-decay schedule (or more precisely, its continuous counterpart, which is the exponentially decaying schedule), and compare this with the polynomially decaying stepsizes. In particular, we consider the use of:

$$\eta_t = \frac{\eta_0}{1+b\cdot t} \quad (6.5) \qquad \eta_t = \frac{\eta_0}{1+b\sqrt{t}} \quad (6.6) \qquad \eta_t = \eta_0\cdot\exp\left(-b\cdot t\right). \quad (6.7)$$

Where, we perform a systematic grid search on the parameters $\eta_0$ and $b$. In the section below, we consider a real world non-convex optimization problem of training a residual network on the cifar-10 dataset, with an aim to illustrate the practical implications of the results described in the chapter. Complete details of the setup are given in Appendix D.5.

[2]Learning rate more than $2/R^2$ will make the algorithm diverge.

### *6.5.1 Non-Convex Optimization: Training a Residual Net on cifar-10*

We consider training a $44-$layer deep residual network [46] with pre-activation blocks [47] (dubbed preresnet-44) on cifar-10 dataset. The code for implementing the network can be found here [3]. For all experiments, we use Nesterov's momentum [85] of 0.9 as implemented in pytorch, batchsize 128, 100 training epochs, $\ell_2$ regularization of 0.0005.

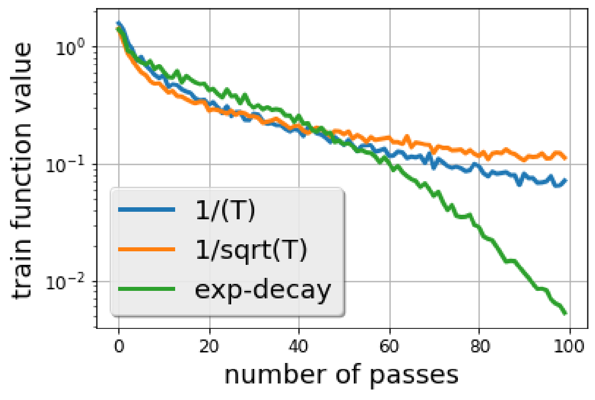
Our experiments are based on grid searching for the best learning rate decay scheme on the parametric family of learning rate schemes described above (6.5),(6.6),(6.7); all grid searches are performed on a separate validation set (obtained by setting aside one-tenth of the training dataset), extended appropriately when searching for parameters and with models trained on the remaining 45000 samples. For presenting the final numbers in the plots/tables, we employ the best hyperparameters from the validation stage and train it on the entire $50,000$ samples and average results run with 10 different random seeds. The parameters for grid searches and other details are presented in Appendix D.5.

**Comparison between different schemes**: Figure 6.2 and Table 6.2 present a comparison of the performance of the three schemes (6.5)-(6.7). These results demonstrate that the exponential scheme convincingly outperforms the polynomial step-size schemes.

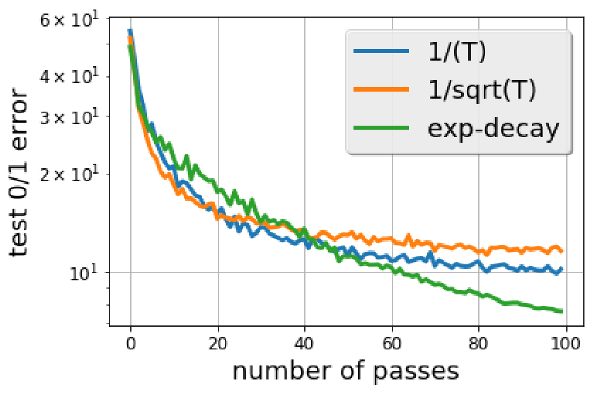
| Decay Scheme | Train Function Value | Test 0/1 error |
|---|---|---|
| $O(1/t)$ (equation (6.5)) | $0.0713 \pm 0.015$ | $10.20 \pm 0.7\%$ |
| $O(1/\sqrt{t})$ (equation (6.6)) | $0.1119 \pm 0.036$ | $11.6 \pm 0.67\%$ |
| $\exp(-t)$ (equation (6.7)) | $\mathbf{0.0053 \pm 0.0015}$ | $\mathbf{7.58 \pm 0.21}\%$ |

Table 6.2: Comparing train cross entropy and test 0/1 error of various learning rate decay schemes on the cifar-10 classification task.

[3]`https://github.com/D-X-Y/ResNeXt-DenseNet`

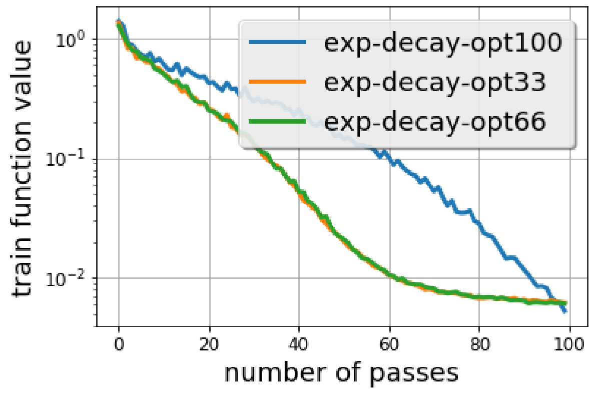


(a) Train Function Value

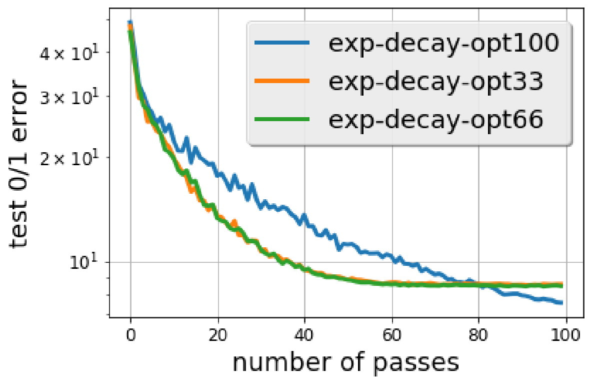


(b) Test 0/1 Error

Figure 6.2: Plots comparing the three decay schemes (two polynomial) (6.5), (6.6), (and one exponential) (6.7) scheme on the cifar-10 classification problem.



(a) Train Function Value



(b) Test 0/1 Error

Figure 6.3: Plots comparing exponential decay scheme (equation (6.7)), with parameters optimized for 33, 66 and 100 epochs on the cifar-10 classification problem.

**Hyperparameter selection using truncated runs**: Figure 6.3 and Tables 6.3 and 6.4 present a comparison of the performance of three exponential decay schemes each of which is tuned to achieve the best performance at 33, 66 and 100 epochs respectively. The key point to note is that best performing hyperparameters at 33 and 66 epochs are not the best performing at 100 epochs (which is made stark from the perspective of the validation error). This demonstrates that hyper parameter selection using truncated runs, (for e.g., in hyperband [70]) would soundly benefit from a round of rethinking.

| Decay Scheme | Train FVal @33 | Train FVal @66 | Train FVal @100 |
|---|---|---|---|
| $\exp(-t)$ [optimized for 33 epochs] (eqn (6.7)) | $\mathbf{0.098 \pm 0.006}$ | $\mathbf{0.0086 \pm 0.002}$ | $\mathbf{0.0062 \pm 0.0015}$ |
| $\exp(-t)$ [optimized for 66 epochs] (eqn (6.7)) | $\mathbf{0.107 \pm 0.012}$ | $\mathbf{0.0088 \pm 0.0014}$ | $\mathbf{0.0061 \pm 0.0011}$ |
| $\exp(-t)$ [optimized for 100 epochs] (eqn (6.7)) | $0.3 \pm 0.06$ | $0.071 \pm 0.017$ | $\mathbf{0.0053 \pm 0.0016}$ |

Table 6.3: Comparing training function value of models obtained by optimizing the exponential decay scheme with end times of 33/66/100 epochs on the cifar-10 classification task.

| Decay Scheme | Test 0/1 @33 | Test 0/1 @66 | Test 0/1 @100 |
|---|---|---|---|
| $\exp(-t)$ [optimized for 33 epochs] (eqn (6.7)) | $\mathbf{10.36 \pm 0.235\%}$ | $\mathbf{8.6 \pm 0.26\%}$ | $8.57 \pm 0.25\%$ |
| $\exp(-t)$ [optimized for 66 epochs] (eqn (6.7)) | $\mathbf{10.51 \pm 0.45\%}$ | $\mathbf{8.51 \pm 0.13\%}$ | $8.46 \pm 0.19\%$ |
| $\exp(-t)$ [optimized for 100 epochs] (eqn (6.7)) | $14.42 \pm 1.47\%$ | $9.8 \pm 0.66\%$ | $\mathbf{7.58 \pm 0.21\%}$ |

Table 6.4: Comparing test 0/1 error of models obtained by optimizing the exponential decay scheme with end times of 33/66/100 epochs on the cifar-10 classification task.

## 6.6 Conclusions and Discussion

The main contribution of this chapter shows that the behavior of SGD's final iterate for least squares regression is much more nuanced than what has been indicated by prior efforts that have primarily considered non-smooth stochastic convex optimization. The results of this chapter point out the striking limitations of polynomially decaying stepsizes on SGD's final iterate, as well as sheds light on the effectiveness of starkly different schemes based on a Step Decay schedule. Somewhat coincidentally, practical implementations for certain classes of stochastic optimization do return the final iterate of SGD with step decay schedule − this connection does merit an understanding through future work.

| | Assumptions | Minimax rate | Rate w/ Final iterate using best poly-decay | Rate w/ Final iterate using Step Decay |
|---|---|---|---|---|
| General convex functions | $\mathbb{E}\left[\left\|\widehat{\nabla} f\right\|^2\right] \leq G^2$ Diam (ConstraintSet) $\leq D$ | $\frac{GD}{\sqrt{T}}$ | $\Theta\left(\frac{GD}{\sqrt{T}} \cdot \log T\right)$ [112, 43] | – |
| Non-strongly convex least squares regression | Eq. (6.3) | $\frac{\sigma^2 d}{T}$ | $\Omega\left(\frac{\sigma^2 d}{T} \cdot \frac{\sqrt{T}}{\log T}\right)$ (This chapter - Theorem 18) | $\mathcal{O}\left(\frac{\sigma^2 d}{T} \cdot \log T\right)$ (This chapter - Theorem 19) |
| General strongly convex functions | $\mathbb{E}\left[\left\|\widehat{\nabla} f\right\|^2\right] \leq G^2$ $\nabla^2 f \succeq \mu \mathbf{I}$ | $\frac{G^2}{\mu T}$ | $\Theta\left(\frac{G^2}{\mu T} \cdot \log T\right)$ [112, 43] | – |
| Strongly convex least squares regression | Eq. (6.3) $\nabla^2 f \succeq \mu \mathbf{I}$ | $\frac{\sigma^2 d}{T}$ | $\Omega\left(\frac{\sigma^2 d}{T} \cdot \kappa\right)$ (This chapter - Theorem 18) | $\mathcal{O}\left(\frac{\sigma^2 d}{T} \cdot \log T\right)$ (This chapter - Theorem 19) |

Table 6.1: Comparison of sub-optimality for *final* iterate of SGD (i.e., $\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*)$) for various stochastic convex optimization settings. This chapter's focus is on SGD's final iterate for streaming least squares regression. The minimax rate refers to the best possible worst case rate with access to stochastic gradients (typically achieved with iterate averaging methods [94, 26, 97]); the red shows the multiplicative factor increase (over the minimax rate) using the final iterate, under two different learning rate schedules - the polynomial decay and the step decay (refer to Algorithm 7). Polynomial decay schedules are of the form $\eta_t \propto 1/t^\alpha$ (for appropriate $\alpha \in [0.5, 1]$). For the general convex cases below, the final iterate with a polynomial decay scheme is off minimax rates by a $\log T$ factor (in an anytime/limiting sense) [112]. Here $\widehat{\nabla} f, \nabla f = \mathbb{E}\left[\widehat{\nabla} f\right], \nabla^2 f$ denotes the stochastic gradient, gradient and the Hessian of the function $f$. With regards to least squares, we assume equation (6.3), following recent efforts [7, 25, 55]. While polynomially decaying stepsizes are nearly minimax optimal for general (strongly) convex functions, this chapter indicates they are highly suboptimal on the final iterate for least squares. The geometrically decaying Step Decay schedule (Algorithm 7) provides marked improvements over any polynomial decay scheme on the final iterate for least squares. For simplicity of presentation, the results for least squares regression do not show dependence on initial error. See Theorems 18 and 19 for precise statements (and [83, 112, 43] for precise statements of the general case).

# Chapter 7
# DISCUSSION

This thesis presents a framework that lends itself towards developing an improved understanding of SGD tailored towards its use in the current era of large scale machine learning and deep learning. We will now discuss some important issues and insights stemming from this line of work.

## 7.1 SGD beyond least squares regression

Despite being corroborated by this thesis' experimental results as well as results in the deep learning literature [114, 79, 40, 110, 75], an obvious means to improve on this thesis' results is through considering extensions towards broader problem classes that go beyond least squares regression (say, towards classes of smooth stochastic convex/non-convex optimization). Progress on this front is conditioned on introducing new tools to analyze the behavior of SGD as applied towards broader function classes, or, alternatively, through a coming up with a reduction based approach that attempts to understand the behavior of SGD using techniques based on linear operators.

## 7.2 Incorporating Second Order Information

The use of sampled second order (Hessian) information in conjunction with stochastic gradients possesses potential advantages mirroring their behavior in deterministic optimization, where, under certain regularity conditions [82], one can expect super-linear (or faster) rates of convergence [14, chap. 9]. However, characterizing the realistic benefits of stochastic second order methods (for the stochastic approximation problem) is non-trivial despite recent work attempting to address these specific issues [7], for reasons described below:

Chapter 4 presented a framework and thought experiment to formalize improvements offered by using acceleration based procedures in conjunction with sampled stochastic gradients (i.e. in the context of inexact first order optimization). One primary takeaway is that the expected improvements offered by accelerating SGD depends heavily on certain distributional properties of the problem, and, that, more importantly, there exists classes of input distributions where the behavior of SGD (in an information theoretic sense) is unimprovable. The behavior of a stochastic second order method that relies on the use of sampled stochastic gradients and hessians naturally is plagued by this worst-case construction (which is information-theoretic) and thus requires a more fine-grained approach towards examining the benefits of stochastic first and second order information, whilst accounting for the specific cases described in chapter 4.

## *7.3 Stochastic Gradient Methods With Higher Order Smoothness*

Extending this thesis' results towards minimizing smooth objectives satisfying notions of Hölder continuous derivatives (with/without uniform convexity) with a stochastic gradient based method is a topic of interesting future direction. This problem has seen an exciting line of work (see [41] and references therein), for results in the deterministic case; results in the stochastic case, in contrast have seen far fewer developments and present exciting avenues for future work, both with and without use of extrapolation style machinery.

## *7.4 Stochastic Approximation With Sparsity Inducing Norms*

$\ell_1-$regularized risk minimization is a mature area with plenty of applications in modern machine learning applications, particularly in high-dimensional statistical learning. Indeed, there has been plenty of efforts directed at solving the offline (empirical) risk minimization problem focusing on co-ordinate descent [106], and in particular, issues involving speeding up optimization [105], parallelization [15, 104] amongst other aspects. There has been progress in the use of multiplicative weight styled procedures with stochastic gradient based methods [52]( with "slow" rates of $\mathcal{O}(1/\sqrt{T})$). Under conditions based on strong convexity of the risk, fast ($\mathcal{O}(1/T)$) rates were indicated by [19, Theorem 3.3]. For lasso style procedures, minimax rates upto an extra strong convexity parameter [128] was achieved by the polynomial time stochastic optimization algorithm of [17]. In the $\ell_0$ setting, the situation has seen more of recent developments, wherein, the work of [36] presents a stochastic optimization methods that achieves optimal rates (as admissible by a polynomial time algorithm) using a successive averaging procedure. It is natural to examine issues examined by this thesis (including parallelization, acceleration and stepsize schemes) for stochastic approximation with $\ell_1$, or $\ell_0$ based regularization (with appropriate technical conditions) to achieve optimal rates of convergence.

# BIBLIOGRAPHY

[1] Alekh Agarwal, Peter L. Bartlett, Pradeep Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 2012. 1, 12, 24, 27, 57, 86, 87

[2] Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Symposium on Theory of Computing (STOC)*, 2017. 10, 24, 43, 53, 79

[3] Zeyuan Allen-Zhu. How to make the gradients small stochastically. In *Neural Information Processing Systems (NeurIPS)*, 2018. 87

[4] Dan Anbar. On optimal estimation methods using stochastic approximation procedures. *The Annals of Statistics*, 1971. 8, 13, 26, 51, 86

[5] Francis Bach and Eric Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Neural Information Processing Systems (NeurIPS) 24*, 2011. 12, 24, 38, 50, 54, 58, 86

[6] Francis R. Bach. Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. *In Journal of Machine Learning Research (JMLR)*, volume 15, 2014. 50, 86

[7] Francis R. Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). In *Neural Information Processing Systems (NeurIPS) 26*, 2013. 12, 26, 28, 31, 38, 50, 52, 54, 82, 83, 85, 86, 88, 89, 91, 96, 97, 119, 146

[8] Albert Benveniste, Michel Metivier, and Pierre Priouret. *Adaptive Algorithms and Stochastic Approximations*. Springer texts in Stochastic Modelling and Applied Probability, 1990. 87

[9] B. Bharath and V. S. Borkar. Stochastic approximation algorithms: overview and recent trends. *Sādhanā*, 1999. 87

[10] Rajendra Bhatia. *Positive Definite Matrices*. Princeton Series in Applied Mathematics. Princeton University Press, 2007. 129

[11] Vivek Borkar. *Stochastic approximation*. Cambridge Books, 2008. 87

[12] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Neural Information Processing Systems (NeurIPS) 20*, 2007. 1, 13, 15, 23, 42, 63, 81

[13] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 2018. 24, 28

[14] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 97

[15] Joseph K. Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel coordinate descent for l1-regularized loss minimization. In *International Conference on Machine Learning (ICML)*, 2011. 16, 25, 98

[16] Sébastien Bubeck. Theory of convex optimization for machine learning. *Foundations and Trends in Machine Learning*, 8 (3-4), 2015. 82, 87, 91

[17] Florentina Bunea, Alexandre B. Tsybakov, and Marten H. Wegkamp. Aggregation for gaussian regression. *Annals of Statistics*, 35(4), 2007. 99

[18] Louis Augustin Cauchy. Méthode générale pour la résolution des systémes d'équations simultanees. *C. R. Acad. Sci. Paris*, 1847. 9, 23, 24, 44, 63, 77

[19] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006. 99

[20] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. In *International Conference on Learning Representations (ICLR)*, 2017. 79, 80

[21] Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *Neural Information Processing Systems (NeurIPS) 24*, 2011. 25, 26

[22] Alexandre d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008. 47, 50, 61, 78

[23] Aaron Defazio. A simple practical accelerated method for finite sums. In *Neural Information Processing Systems (NeurIPS) 29*, 2016. 10, 24, 79

[24] Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Neural Information Processing Systems (NeurIPS) 27*, 2014. 10, 24, 79

[25] Alexandre Défossez and Francis R. Bach. Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Artifical Intelligence and Statistics (AISTATS)*, 2015. 21, 22, 23, 24, 25, 27, 28, 30, 31, 37, 40, 50, 52, 54, 82, 83, 86, 89, 96, 119

[26] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research (JMLR)*, volume 13, 2012. 25, 26, 28, 82, 87, 96

[27] Olivier Devolder, Franccois Glineur, and Yurii E. Nesterov. First-order methods with inexact oracle: the strongly convex case. *CORE Discussion Papers 2013016*, 2013. 47, 50, 61

[28] Olivier Devolder, Franccois Glineur, and Yurii E. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146:37–75, 2014. 47, 50, 61, 78

[29] Aymeric Dieuleveut and Francis Bach. Non-parametric stochastic approximation with large step sizes. *The Annals of Statistics*, 2015. 25, 31, 50, 52, 86

[30] Aymeric Dieuleveut, Nicolas Flammarion, and Francis R. Bach. Harder, better, faster, stronger convergence rates for least-squares regression. *Journal of Machine Learning Research (JMLR)*, 18, 2017. 52, 78, 87

[31] John C. Duchi, Sorathan Chaturapruek, and Christopher Ré. Asynchronous stochastic convex optimization. In *Neural Information Processing Systems (NeurIPS)*, 2016. 26

[32] John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011. 79, 81

[33] Vaclav Fabian. Asymptotically efficient stochastic approximation; the RM case. *Annals of Statistics*, 1(3), 1973. 8, 13, 26, 51

[34] Roy Frostig, Rong Ge, Sham Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *International Conference on Machine Learning (ICML)*, 2015. 10, 24, 53, 79

[35] Roy Frostig, Rong Ge, Sham M. Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In *Conference on Learning Theory (COLT)*, 2015. 12, 13, 24, 25, 26, 27, 28, 32, 38, 43, 45, 48, 50, 52, 53, 57, 58, 60, 79, 86, 119, 146

[36] Pierre Gaillard and Olivier Wintenberger. Sparse accelerated exponential weights. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017)*, volume 54, pages 75–82. PMLR, 2017. 99

[37] Rong Ge, Sham Kakade, Rahul Kidambi, and Praneeth Netrapalli. The step decay schedule: A near optimal, geometrically decaying learning rate procedure. *pre-print*, 2019. 81, 92

[38] Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework. *SIAM Journal on Optimization*, 2012. 52, 78, 87

[39] Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 2013. 52, 78, 87

[40] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: training imagenet in 1 hour. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 21, 31, 97

[41] G. Grapiglia and Y. Nesterov. Tensor methods for minimizing functions with hölder continuous higher-order derivatives. *Corr*, 2018. 98

[42] Anne Greenbaum. Behavior of slightly perturbed lanczos and conjugate-gradient recurrences. *Linear Algebra and its Applications*, 1989. 47, 51, 61, 78

[43] Nicholas J. A. Harvey, Christopher Liaw, Yaniv Plan, and Sikander Randhawa. Tight analyses for non-smooth stochastic gradient descent. *CoRR*, 2018. 82, 87, 96

[44] Babak Hassibi, Ali H. Sayed, and Thomas Kailath. $h^\infty$ optimality of the lms algorithm. *IEEE Transactions on Signal Processing*, 1996. 11, 51

[45] Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *Journal of Machine Learning Research (JMLR)*, volume 15, 2014. 87

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 75, 93

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV (4)*, Lecture Notes in Computer Science, pages 630–645. Springer, 2016. 65, 75, 93

[48] Magnus R. Hestenes and Eduard Stiefel. Methods of conjuate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 1952. 9, 45, 50

[49] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 72

[50] Daniel J. Hsu, Sham M. Kakade, and Tong Zhang. Random design analysis of ridge regression. *Foundations of Computational Mathematics*, 14(3):569–600, 2014. 48, 55

[51] Chonghai Hu, James T. Kwok, and Weike Pan. Accelerated gradient methods for stochastic optimization and online learning. In *Neural Information Processing Systems (NeurIPS)*, 2009. 52

[52] Kivinen J. and Warmuth M.K. Additive versus exponentiated gradient updates for linear prediction. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 1995. 98

[53] Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, Venkata Krishna Pillutla, and Aaron Sidford. A markov chain theory approach to characterizing the minimax optimality of stochastic gradient descent (for least squares). In *FSTTCS (invited paper)*, 2017. 22, 83, 119, 122, 126, 223

[54] Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Accelerating stochastic gradient descent. In *Conference on Learning Theory (COLT)*, 2018. 24, 25, 26, 64, 67, 68, 70, 78, 79, 80, 83, 86, 126, 212, 213

[55] Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Parallelizing stochastic approximation through mini-batching and tail-averaging. *Journal of Machine Learning Research (JMLR)*, 2018. iv, 28, 43, 45, 46, 48, 49, 50, 52, 54, 56, 57, 58, 60, 68, 82, 83, 86, 88, 89, 91, 96, 146, 216, 217, 221

[56] Prateek Jain, Dheeraj Nagaraj, and Praneeth Netrapalli. Making the last iterate of sgd information theoretically optimal. In *Conference on Learning Theory (COLT)*, 2019. 82, 87, 91

[57] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Neural Information Processing Systems (NeurIPS) 26*, 2013. 10, 24, 79, 81

[58] S. Kaczmarz. Angenaherte auflosung von systemen linearer gleichungen. *Bull. Acad. Polon*, 1937. 2, 42, 49

[59] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 79

[60] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009. 75

[61] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NeurIPS)*, 2012. 1

[62] Harold J. Kushner and Dean S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* Springer-Verlag, 1978. 8, 11, 26, 44, 51, 52, 87, 89

[63] Harold J. Kushner and G. Yin. Asymptotic properties of distributed and communicating stochastic approximation algorithms. *SIAM Journal on Control and Optimization*, 25(5):1266–1290, 1987. 23

[64] Harold J. Kushner and G. Yin. Stochastic approximation and recursive algorithms and applications. *Springer-Verlag*, 2003. 8, 13, 23, 26, 52, 87

[65] Simon Lacoste-Julien, Mark W. Schmidt, and Francis R. Bach. A simpler approach to obtaining an o(1/t) convergence rate for the projected stochastic subgradient method. *CoRR*, 2012. 82, 87, 91

[66] Tze Leung Lai. Stochastic approximation: invited paper, 2003. 87

[67] Guanghui Lan. An optimal method for stochastic composite optimization. *Math. Programming*, 133, 2012. 52

[68] Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *Mathematical Programming*, 2017. 53

[69] Erich L. Lehmann and George Casella. *Theory of Point Estimation*. Springer Texts in Statistics. Springer, 1998. 8, 11, 20, 26, 44, 52, 57, 86

[70] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017. 95

[71] Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J. Smola. Efficient mini-batch training for stochastic optimization. In *Knowledge Discovery and Data Mining (KDD)*, 2014. 16, 25, 26

[72] Hongzhou Lin, Julien Mairal, and Zaïd Harchaoui. A universal catalyst for first-order optimization. In *Neural Information Processing Systems (NeurIPS)*, 2015. 10, 24, 53, 79

[73] Lennart Ljung, Georg Pflug, and Harro Walk. *Stochastic Approximation and Optimization of Random Systems*. Birkhauser Verlag, Basel, Switzerland, Switzerland, 1992. 87

[74] Nicolas Loizou and Peter Richtárik. Linearly convergent stochastic heavy ball method for minimizing generalization error. In *NeurIPS workshop on Optimization for Machine Learning*, 2017. 79

[75] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017. 97

[76] Gideon Mann, Ryan T. McDonald, Mehryar Mohri, Nathan Silberman, and Dan Walker. Efficient large-scale distributed training of conditional maximum entropy models. In *Neural Information Processing Systems (NeurIPS) 22*, 2009. 16, 25, 36

[77] James Martens. Deep learning via hessian-free optimization. In *International conference on Machine Learning (ICML)*, 2010. 72

[78] James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on Machine Learning (ICML)*, 2015. 72

[79] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing lstm language models. In *International Conference on Learning Representations (ICLR)*, 2018. 21, 30, 97

[80] Jin-Ichi Nagumo and Atsuhiko Noda. A learning method for system identification. *IEEE Transactions on Automatic Control*, 1967. 86

[81] Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. *Mathematical Programming*, volume 155, 2016. 24, 50, 86

[82] A. S. Nemirovsky and Y. Nesterov. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, 1994. 97

[83] Arkadi S. Nemirovsky and David B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley, 1983. 1, 9, 12, 23, 43, 51, 52, 57, 86, 87, 96

[84] Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming Series B*, 2012. 63

[85] Yurii E. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Doklady AN SSSR*, 269, 1983. i, 2, 9, 23, 45, 63, 64, 66, 77, 78, 93, 240

[86] Yurii E. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004. 9, 12, 45, 63, 64, 77

[87] Yurii E. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012. 48, 55, 64, 78

[88] Gergely Neu and Lorenzo Rosasco. Iterate averaging as regularization for stochastic gradient descent. In *Conference on Learning Theory (COLT)*, 2018. 83, 86

[89] Behnam Neyshabur, Ruslan Salakhutdinov, and Nathan Srebro. Path-sgd: Path-normalized optimization in deep neural networks. In *Neural Information Processing Systems (NeurIPS)*, 2015. 80

[90] Feng Niu, Benjamin Recht, Christopher Re, and Stephen J. Wright. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Neural Information Processing Systems (NeurIPS) 24*, 2011. 16, 25, 26

[91] Christopher C. Paige. The computation of eigenvalues and eigenvectors of very large sparse matrices. *PhD Thesis, University of London*, 1971. 47, 51, 61, 78

[92] Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4, 1964. i, 2, 9, 23, 45, 50, 62, 63, 64, 66, 77, 78, 79

[93] Boris T. Polyak. *Introduction to Optimization*. Optimization Software, 1987. 47, 50, 61, 78

[94] Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J Control Optim*, volume 30, 1992. , i, 1, 3, 11, 19, 20, 21, 25, 26, 31, 42, 44, 50, 51, 52, 56, 57, 60, 82, 83, 86, 89, 91, 96

[95] John G. Proakis. Channel identification for high speed digital communications. *IEEE Transactions on Automatic Control*, 1974. 47, 51, 61, 78, 82, 86

[96] Maxim Raginsky and Alexander Rakhlin. Information-based complexity, feedback and dynamics in convex programming. *IEEE Transactions on Information Theory*, 2011. 1, 57, 87

[97] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *International Conference on Machine Learning (ICML)*, 2012. 10, 19, 79, 82, 87, 91, 96

[98] Sashank Reddi, Manzil Zaheer, Suvrit Sra, Barnabas Poczos, Francis Bach, Ruslan Salakhutdinov, and Alexander Smola. A generic approach for escaping saddle points. In *Artifical Intelligence and Statistics (AISTATS)*, 2017. 72

[99] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Stat.*, vol. 22, 1951. , i, 1, 3, 10, 11, 18, 23, 42, 50, 56, 60, 63, 77, 81, 86, 89

[100] Jonathan Rosenblatt and Boaz Nadler. On the optimality of averaging in distributed statistical learning. *Information and Inference: a journal of the IMA*, 2016. 25, 26, 36

[101] Nicolas Le Roux, Mark Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. In *Neural Information Processing Systems (NeurIPS) 25*, 2012. 10, 24, 79

[102] Sumit Roy and John J. Shynk. Analysis of the momentum lms algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1990. 47, 51, 61, 78, 82, 86

[103] David Ruppert. Efficient estimations from a slowly convergent robbins-monro process. *Tech. Report, ORIE, Cornell University*, 1988. , i, 1, 11, 19, 20, 25, 42, 44, 50, 60, 82, 83, 86, 89, 91

[104] Chad Scherrer, Mahantesh Halappanavar, Ambuj Tewari, and David Haglin. Scaling up coordinate descent algorithms for large $l_1$ regularization problems. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1407–1414. Omnipress, 2012. 98

[105] Chad Scherrer, Ambuj Tewari, Mahantesh Halappanavar, and David Haglin. Feature clustering for accelerating parallel coordinate descent. In *Advances in Neural Information Processing Systems 25*, pages 28–36, 2012. 98

[106] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for $l_1$ regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, June 2011. 98

[107] Shai Shalev-Shwartz and Tong Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *Neural Information Processing Systems (NeurIPS 26)*, 2013. 24, 25

[108] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research (JMLR)*, 2013. 10, 24, 79

[109] Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *International Conference on Machine Learning (ICML)*, 2014. 10, 53, 79

[110] Christopher J. Shallue, Jaehoon Lee, Joe Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E. Dahl. Measuring the effects of data parallelism on neural network training. *CoRR*, abs/1811.03600, 2018. 97

[111] Ohad Shamir. Open problem: Is averaging needed for strongly convex stochastic gradient descent? In *Conference on Learning Theory (COLT)*, 2012. 82

[112] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning (ICML)*, 2013. 82, 87, 91, 96

[113] Rajesh Sharma, William A. Sethares, and James A. Bucklew. Analysis of momentum adaptive filtering algorithms. *IEEE Transactions on Signal Processing*, 1998. 47, 51, 61, 78, 86

[114] Samuel L Smith, Pieter-Jan Kindermans, and Quoc V Le. Don't decay the learning rate, increase the batch size. In *International Conference on Learning Representations (ICLR)*, 2018. 21, 33, 34, 36, 97

[115] Thomas Strohmer and Roman Vershynin. A randomized kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 2007. 2, 42, 49

[116] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on Machine Learning (ICML)*, pages 1139–1147, 2013. 63, 64, 72, 73, 79

[117] Martin Takác, Avleen Singh Bijral, Peter Richtárik, and Nati Srebro. Mini-batch primal and dual methods for SVMs. In *International Conference on Machine Learning (ICML)*, volume 28, 2013. 25

[118] Martin Takác, Peter Richtárik, and Nati Srebro. Distributed mini-batch sdca. *Journal of Machine Learning Research (JMLR)*, 2018. 25

[119] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 2012. 79

[120] Aad W. van der Vaart. *Asymptotic Statistics*. Cambridge University Publishers, 2000. 1, 8, 11, 20, 44, 52, 57, 89, 91

[121] Bernard Widrow and Marcian E Hoff. *Adaptive switching circuits*. Defense Technical Information Center, 1960. 82, 86

[122] Bernard Widrow and Samuel D. Stearns. *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985. 11, 51, 82, 86

[123] Ashia C. Wilson, Benjamin Recht, and Michael I. Jordan. A lyapunov analysis of momentum methods in optimization. *CoRR*, abs/1611.02635, 2016. 59

[124] Blake Woodworth and Nathan Srebro. Tight complexity bounds for optimizing composite objectives. In *Neural Information Processing Systems (NeurIPS)*, 2016. 10, 53

[125] Kun Yuan, Bicheng Ying, and Ali H. Sayed. On the influence of momentum acceleration on online learning. *Journal of Machine Learning Research (JMLR)*, volume 17, 2016. 47, 51, 61

[126] Jian Zhang and Ioannis Mitliagkas. Yellowfin and the art of momentum tuning. In *SysML conference*, 2019. 80

[127] Yuchen Zhang, John C. Duchi, and Martin Wainwright. Divide and conquer ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research (JMLR)*, volume 16, 2015. 25, 36

[128] Yuchen Zhang, Martin J. Wainwright, and Michael I. Jordan. Optimal prediction for sparse linear models? lower bounds for coordinate-separable m-estimators. *Electronic Journal of Statistics*, 2015. 99

[129] Yuchen Zhang and Lin Xiao. Disco: Distributed optimization for self-concordant empirical loss. In *International Conference on Machine Learning (ICML)*, 2015. 16

[130] Martin A. Zinkevich, Alex Smola, Markus Weimer, and Lihong Li. Parallelized stochastic gradient descent. In *Neural Information Processing Systems (NeurIPS) 24*, 2011. 16, 36

# Appendix A

# APPENDIX: TAIL-AVERAGED SGD: PARALLELIZATION VIA MINI-BATCHING, MODEL AVERAGING AND BATCHSIZE DOUBLING

We begin with a note on the organization:

- Section A.1 introduces notations necessary for the rest of the appendix.
- Section A.2 derives the mini-batch SGD update and provides the bias-variance decomposition and reasons about its implication in bounding the generalization error.
- Section A.3 provides lemmas that are used to bound the bias error.
- Section A.4 provides lemmas that are used to bound the variance error.
- Section A.5 uses the results of the previous sections to obtain the main results of this chapter.

## *A.1 Notations*

We begin by introducing the centered iterate $\boldsymbol{\eta}_t$ i.e.:

$$\boldsymbol{\eta}_t \overset{\text{def}}{=} \mathbf{w}_t - \mathbf{w}^*.$$

In a manner similar to $\mathbf{w}_t$, the tail-averaged iterate $\overline{\mathbf{w}}_{t,N}$ is associated with its corresponding centered estimate $\bar{\boldsymbol{\eta}}_{t,N} \overset{\text{def}}{=} \overline{\mathbf{w}}_{t,N} - \mathbf{w}^* = \frac{1}{N}\sum_{s=t}^{t+N-1}(\mathbf{w}_s - \mathbf{w}^*) = \frac{1}{N}\sum_{s=t}^{t+N-1}\boldsymbol{\eta}_s$. Next, let $\boldsymbol{\Phi}_t$ denote the expected covariance of the centered estimate $\boldsymbol{\eta}_t$, i.e.

$$\boldsymbol{\Phi}_t \overset{\text{def}}{=} \mathbb{E}\left[\boldsymbol{\eta}_t \otimes \boldsymbol{\eta}_t\right],$$

and in a similar way as the final iterate $\mathbf{w}_t$, the tail-averaged estimate $\overline{\mathbf{w}}_{t,N}$ is associated with its expected covariance, i.e. $\bar{\mathbf{\Phi}}_{t,N} \overset{\text{def}}{=} \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{t,N} \otimes \bar{\boldsymbol{\eta}}_{t,N}\right]$.

## A.2 Mini-Batch Tail-Averaged SGD: Bias-Variance Decomposition

In section A.2.1, we derive the basic recursion governing the evolution of the iterates $\mathbf{w}_t$ and the tail-averaged iterate $\overline{\mathbf{w}}_{s+1,N}$. In section A.2.2 we provide the bias-variance decomposition of the final iterate. In section A.2.3, we provide the bias-variance decomposition of the tail-averaged iterate.

### *A.2.1 The basic recursion*

At each iteration $t$ of Algorithm 1, we are provided with $b$ fresh samples $\{(\mathbf{x}_{ti}, y_{ti})\}_{i=1}^{b}$ drawn i.i.d. from the distribution $\mathcal{D}$. We start by recounting the mini-batch gradient descent update rule that allows us to move from iterate $\mathbf{w}_{t-1}$ to $\mathbf{w}_t$:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \frac{\gamma}{b}\sum_{i=1}^{b}(\langle \mathbf{w}_{t-1}, \mathbf{x}_{ti}\rangle - y_{ti})\mathbf{x}_{ti}$$

where, $0 < \gamma < \gamma_{b,\max}$ is the constant step size that is set to a value less than the maximum allowed learning rate $\gamma_{b,\max}$. We also recount the definition of $\overline{\mathbf{w}}_{t,N}$ which is the iterate obtained by averaging for $N$ iterations starting from the $t^{th}$ iteration, i.e.,

$$\overline{\mathbf{w}}_{t,N} = \frac{1}{N}\sum_{s=t}^{t+N-1}\mathbf{w}_s$$

Let us first denote the residual error term by $\epsilon_i = y_i - \langle \mathbf{w}^*, \mathbf{x}_i\rangle$. By the first order optimality conditions of $\mathbf{w}^*$, we observe that $\epsilon$ and $\mathbf{x}$ are orthogonal, i.e, $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\epsilon \cdot \mathbf{x}] = 0$. For any estimate $\mathbf{w}$, the excess risk/generalization error can be written as:

$$L(\mathbf{w}) - L(\mathbf{w}^*) = \frac{1}{2}\operatorname{Tr}\Big(\mathbf{H}\cdot(\boldsymbol{\eta}\otimes\boldsymbol{\eta})\Big), \text{ with } \boldsymbol{\eta} = \mathbf{w} - \mathbf{w}^*. \tag{A.1}$$

We now write out the main recursion governing the mini-batch SGD updates in terms of $\boldsymbol{\eta}_{.}$:

$$\boldsymbol{\eta}_t = \Big(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\Big)\boldsymbol{\eta}_{t-1} + \frac{\gamma}{b}\sum_{i=1}^{b}\epsilon_{ti}\mathbf{x}_{ti}$$

$$= \Big(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\Big)\boldsymbol{\eta}_{t-1} + \frac{\gamma}{b}\sum_{i=1}^{b}\boldsymbol{\xi}_{ti}$$

$$= \mathbf{P}_{tb}\boldsymbol{\eta}_{t-1} + \gamma\boldsymbol{\zeta}_{tb} \tag{A.2}$$

Where, $\mathbf{P}_{tb} \overset{\text{def}}{=} \Big(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\Big)$ and $\boldsymbol{\zeta}_{tb} \overset{\text{def}}{=} \frac{1}{b}\sum_{i=1}^{b}\boldsymbol{\xi}_{ti} = \frac{1}{b}\sum_{i=1}^{b}\epsilon_{ti}\mathbf{x}_{ti}$. Equation (A.2) automatically brings out the "operator" view of analyzing the (expected) covariance of the centered estimate $\boldsymbol{\Phi}_t = \mathbb{E}\left[\boldsymbol{\eta}_t\otimes\boldsymbol{\eta}_t\right]$ to provide an estimate of the generalization error. We now note the following about the covariance of $\boldsymbol{\zeta}_{tb}$:

$$\mathbb{E}[\boldsymbol{\zeta}_{tb}\otimes\boldsymbol{\zeta}_{t'b}] = \frac{1}{b^2}\sum_{i,j}\mathbb{E}[\boldsymbol{\xi}_{ti}\otimes\boldsymbol{\xi}_{t'j}]$$

$$= \Big[\frac{1}{b^2}\sum_{i=1}^{b}\mathbb{E}[\boldsymbol{\xi}_{ti}\otimes\boldsymbol{\xi}_{ti}]\Big]\mathbb{1}[t=t'] = \frac{1}{b}\boldsymbol{\Sigma}\;\;\mathbb{1}[t=t'] \tag{A.3}$$

Where, $\mathbb{1}[.]$ is the indicator function, and equals 1 if the argument inside [.] is true and 0 otherwise. We note that the expectation of the cross terms in equation (A.3) is zero owing to independence of the samples $\{\mathbf{x}_{ti}, y_{ti}\}_{i=1}^{b}$ as well as between $\{\mathbf{x}_{ti}, y_{ti}\}_{i=1}^{b}$, $\{\mathbf{x}_{t'i}, y_{t'i}\}_{i=1}^{b}$ $\forall\, t\neq t'$ and owing to the first order optimality conditions. Owing to the invariance of $\boldsymbol{\zeta}_{tb}$ on the iteration $t$, context permitting, we sometimes drop the iteration index $t$ from $\boldsymbol{\zeta}_{tb}$ and simply refer to it as $\boldsymbol{\zeta}_{b}$.

Next we expand out the recurrence (A.2). Let $\mathbf{Q}_{j,t} = (\prod_{k=j}^{t}\mathbf{P}_{kb})^T$ with the convention that $\mathbf{Q}_{t',t} = \mathbf{I}\ \forall\ t' > t$. With this notation we have:

$$\boldsymbol{\eta}_t = \mathbf{P}_{tb}\boldsymbol{\eta}_{t-1} + \gamma\boldsymbol{\zeta}_{tb}$$

$$= \mathbf{P}_{tb}\mathbf{P}_{t-1,b}...\mathbf{P}_{1,b}\boldsymbol{\eta}_0 + \gamma\sum_{j=0}^{t-1}\{\mathbf{P}_{tb}....\mathbf{P}_{t-j+1,b}\}\boldsymbol{\zeta}_{t-j,b}$$

$$= \mathbf{Q}_{1,t}\boldsymbol{\eta}_0 + \gamma\sum_{j=0}^{t-1}\mathbf{Q}_{t-j+1,t}\boldsymbol{\zeta}_{t-j,b}$$

$$= \mathbf{Q}_{1,t}\boldsymbol{\eta}_0 + \gamma\sum_{j=1}^{t}\mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b}$$

$$= \boldsymbol{\eta}_t^{\text{bias}} + \boldsymbol{\eta}_t^{\text{variance}}\ , \tag{A.4}$$

where, we note that

$$\boldsymbol{\eta}_t^{\text{bias}} \stackrel{\text{def}}{=} \mathbf{Q}_{1,t}\boldsymbol{\eta}_0, \tag{A.5}$$

relates to understanding the behavior of SGD on the noiseless problem (i.e. $\boldsymbol{\zeta}_{\cdot,\cdot} = 0$ a.s.) and aims to quantify the dependence on the initial conditions. Further,

$$\boldsymbol{\eta}_t^{\text{variance}} \stackrel{\text{def}}{=} \gamma \sum_{j=1}^{t} \mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b} \tag{A.6}$$

relates to the behavior of SGD when begun at the solution (i.e. $\boldsymbol{\eta}_0 = 0$) and allowing the noise $\boldsymbol{\zeta}_{\cdot,\cdot}$ to drive the process.

Furthermore, considering the tail-averaged iterate obtained by averaging the iterates of the SGD procedure for $N$ iterations starting from a certain number of iterations "$s$", i.e., we examine the quantity $\bar{\boldsymbol{\eta}}_{s+1,N} = \overline{\mathbf{w}}_{s+1,N} - \mathbf{w}^*$, where $\overline{\mathbf{w}}_{s+1,N} = \frac{1}{N}\sum_{t=s+1}^{s+N}\mathbf{w}_t$. We write out the expression for $\bar{\boldsymbol{\eta}}_{s+1,N}$ starting out from equation (A.4):

$$\begin{aligned}
\bar{\boldsymbol{\eta}}_{s+1,N} &= \frac{1}{N}\sum_{t=s+1}^{s+N} \boldsymbol{\eta}_t \\
&= \frac{1}{N}\sum_{t=s+1}^{s+N} \left(\boldsymbol{\eta}_t^{\text{bias}} + \boldsymbol{\eta}_t^{\text{variance}}\right) && \text{(from equation (A.4))} \\
&= \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{bias}} + \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{variance}}.
\end{aligned} \tag{A.7}$$

## *A.2.2 The Final Iterate: Bias-Variance Decomposition*

The behavior of the final iterate is considered to be of great practical interest and we hope to shed light on the behavior of this final iterate and the tradeoffs between the learning rate and batch size. Since the generalization error of any iterate $\mathbf{w}_N$ obtained by running mini-batch SGD with a batch size $b$ for a total of $N$ iterations can be estimated by tracking $\mathbb{E}\left[\boldsymbol{\eta}_N \otimes \boldsymbol{\eta}_N\right]$, where, $\boldsymbol{\eta}_N = \mathbf{w}_N - \mathbf{w}^*$, we provide a simple psd upper bound on the outer product of interest, i.e.:

$$\mathbb{E}\left[\boldsymbol{\eta}_N \otimes \boldsymbol{\eta}_N\right] = \mathbb{E}\left[\left(\boldsymbol{\eta}_N^{\text{bias}} + \boldsymbol{\eta}_N^{\text{variance}}\right) \otimes \left(\boldsymbol{\eta}_N^{\text{bias}} + \boldsymbol{\eta}_N^{\text{variance}}\right)\right] \quad \text{(by substituting equation (A.4))}$$

$$\preceq 2 \cdot \Big( \mathbb{E}\left[ (\boldsymbol{\eta}_N^{\text{bias}} \otimes \boldsymbol{\eta}_N^{\text{bias}}) \right] + \mathbb{E}\left[ (\boldsymbol{\eta}_N^{\text{variance}} \otimes \boldsymbol{\eta}_N^{\text{variance}}) \right] \Big)$$

Using this expression, we now write out the expression for the excess risk of the final iterate:

$$\begin{aligned}
\mathbb{E}\left[L(\mathbf{w}_N)\right] - L(\mathbf{w}^*) &= \frac{1}{2}\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_N \otimes \boldsymbol{\eta}_N\right]\rangle \\
&\leq \frac{1}{2}\langle \mathbf{H}, 2 \cdot \left( \mathbb{E}\left[\boldsymbol{\eta}_N^{\text{bias}} \otimes \boldsymbol{\eta}_N^{\text{bias}}\right] + \mathbb{E}\left[\boldsymbol{\eta}_N^{\text{variance}} \otimes \boldsymbol{\eta}_N^{\text{variance}}\right] \right)\rangle \\
&\leq 2 \cdot \Big( \frac{1}{2}\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_N^{\text{bias}} \otimes \boldsymbol{\eta}_N^{\text{bias}}\right]\rangle + \frac{1}{2}\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_N^{\text{variance}} \otimes \boldsymbol{\eta}_N^{\text{variance}}\right]\rangle \Big) \\
&= 2 \cdot \Big( \left( \mathbb{E}\left[L(\mathbf{w}_N^{\text{bias}})\right] - L(\mathbf{w}^*) \right) + \left( \mathbb{E}\left[L(\mathbf{w}_N^{\text{variance}})\right] - L(\mathbf{w}^*) \right) \Big). \qquad \text{(A.8)}
\end{aligned}$$

### *A.2.3 The Tail-Averaged Iterate: Bias-Variance Decomposition*

Now, considering the fact that the excess risk/generalization error (equation (A.1)) involves tracking $\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right]$, we see that the quantity of interest can be bounded by considering the behavior of SGD on bias and variance sub-problem. In particular, writing out the outerproduct of equation (A.7), we see the following inequality holds through a straightforward application of Cauchy-Shwartz inequality:

$$\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right] \preceq 2 \cdot \left( \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N}^{\text{bias}} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{bias}}\right] + \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N}^{\text{variance}} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{variance}}\right] \right) \qquad \text{(A.9)}$$

The equation above is referred to as the bias-variance decomposition and is well known from previous work on Stochastic Approximation [7, 35, 25]. This implies that an upper bound on the generalization error (equation (A.1)) is:

$$\begin{aligned}
L(\overline{\mathbf{w}}_{s+1,N}) - L(\mathbf{w}^*) &= \frac{1}{2}\langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right]\rangle \\
&\leq \langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N}^{\text{bias}} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{bias}}\right]\rangle + \langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N}^{\text{variance}} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{variance}}\right]\rangle. \qquad \text{(A.10)}
\end{aligned}$$

Here, we adopt the proof approach of [53]. In particular, [53] provide a clean way to simplify the expression corresponding to the tail-averaged iterate. Let us consider $\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right]$ and simplify the resulting expression: in particular,

$$\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right] = \frac{1}{N^2} \sum_{l=s+1}^{s+N} \sum_{k=s+1}^{s+N} \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_k\right]$$

$$\begin{aligned}
&= \frac{1}{N^2}\cdot\left(\sum_{l\geq k}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_k\right]+\sum_{l<k}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_k\right]\right)\\
&\preceq \frac{1}{N^2}\cdot\left(\sum_{l\geq k}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_k\right]+\sum_{l\leq k}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_k\right]\right) \qquad (*)\\
&= \frac{1}{N^2}\cdot\left(\sum_{l\geq k}(\mathbf{I}-\gamma\mathbf{H})^{l-k}\mathbb{E}\left[\boldsymbol{\eta}_k\otimes\boldsymbol{\eta}_k\right]+\sum_{l\leq k}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l}\right) \quad (**)\\
&= \frac{1}{N^2}\cdot\sum_{l\leq k}\left(\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l}+(\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right)\\
&= \frac{1}{N^2}\cdot\sum_{l=s+1}^{s+N}\sum_{k=l}^{s+N}\left(\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l}+(\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right)\\
&= \frac{1}{N^2}\cdot\sum_{l=s+1}^{s+N}\sum_{k=l}^{\infty}\left(\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l}+(\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right)\\
&- \frac{1}{N^2}\cdot\sum_{l=s+1}^{s+N}\sum_{k=s+N+1}^{\infty}\left(\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l}+(\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right)\\
&= \frac{1}{N^2}\cdot\sum_{l=s+1}^{s+N}\left(\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right](\gamma\mathbf{H})^{-1}+(\gamma\mathbf{H})^{-1}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right)\\
&- \frac{1}{N^2}\cdot\sum_{l=s+1}^{s+N}\sum_{k=s+N+1}^{\infty}\left(\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l}+(\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right)
\end{aligned} \tag{A.11}$$

where, $(*)$ is a valid PSD upper bound since we add and subtract the diagonal terms $\{\mathbb{E}\left[\boldsymbol{\eta}_k\otimes\boldsymbol{\eta}_k\right]\}_{k=s+1}^{s+N}$. $(**)$ follows because of the following (assume $l > k$; the other case follows similarly):

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_k\right] &= \mathbb{E}\left[\left(\mathbf{P}_{lb}\boldsymbol{\eta}_{l-1}+\gamma\boldsymbol{\zeta}_{lb}\right)\otimes\boldsymbol{\eta}_k\right]\\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\mathbf{P}_{lb}\boldsymbol{\eta}_{l-1}+\gamma\boldsymbol{\zeta}_{lb}\right)\otimes\boldsymbol{\eta}_k|\mathcal{F}_{l-1}\right]\right]\\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\mathbf{P}_{lb}\boldsymbol{\eta}_{l-1}+\gamma\boldsymbol{\zeta}_{lb}\right)|\mathcal{F}_{l-1}\right]\otimes\boldsymbol{\eta}_k\right]\\
&= (\mathbf{I}-\gamma\mathbf{H})\mathbb{E}\left[\boldsymbol{\eta}_{l-1}\otimes\boldsymbol{\eta}_k\right],
\end{aligned}$$

where, the final equation follows since $\mathbb{E}\left[\mathbf{P}_{lb}|\mathcal{F}_{l-1}\right] = \mathbb{E}\left[\mathbf{I}-\frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{li}\otimes\mathbf{x}_{li}|\mathcal{F}_{l-1}\right] = \mathbf{I}-\gamma\mathbf{H}$ and $\mathbb{E}\left[\boldsymbol{\zeta}_{lb}|\mathcal{F}_{l-1}\right]=0$ from first order optimality conditions. Recursing over $l$ yields the result.

The final line simply follows from summing a (convergent) geometric series.

This implies that the excess risk corresponding to the bias/variance term can be obtained from equation (A.11) by taking an inner product with $\mathbf{H}$, i.e.:

$$\begin{aligned}\langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right]\rangle \leq \frac{1}{N^2} \cdot \sum_{l=s+1}^{s+N} \Big( &\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\gamma\mathbf{H})^{-1} + (\gamma\mathbf{H})^{-1}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\rangle \Big) \\ &- \frac{1}{N^2} \cdot \sum_{l=s+1}^{s+N} \sum_{k=s+N+1}^{\infty} \Big( \langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l} \\ &\qquad + (\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\rangle \Big) \\ &\leq \frac{1}{N^2} \cdot \sum_{l=s+1}^{s+N} \Big( \langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\gamma\mathbf{H})^{-1} + (\gamma\mathbf{H})^{-1}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\rangle \Big) \\ &= \frac{2}{\gamma N^2} \cdot \sum_{l=s+1}^{s+N} \operatorname{Tr}\Big( \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right] \Big) \end{aligned} \tag{A.12}$$

The upper bound on the final line follows because each term within the summation in the second line is negative owing to the following argument. Consider say,

$$\begin{aligned}&\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l} + (\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\rangle \\ &= 2\operatorname{Tr}\left[\mathbf{H}(\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right] \geq 0.\end{aligned}$$

Note that $\mathbf{H}$ and $(\mathbf{I}-\gamma\mathbf{H})$ commute and both are psd, implying that $\mathbf{H}(\mathbf{I}-\gamma\mathbf{H})^{k-l}$ is PSD. Finally, the trace of the product of two PSD matrices is positive with $\mathbf{H}(\mathbf{I}-\gamma\mathbf{H})^{k-l}$ being one of these PSD matrices and $\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]$ being the other, thus yielding the claimed bound in equation (A.12).

This implies that the overall error (through equation (A.1)) can be upperbounded as:

$$\begin{aligned}\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N})\right] - L(\mathbf{w}^*) &= \frac{1}{2} \cdot \langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right]\rangle \\ &\leq \frac{1}{\gamma N^2} \sum_{l=s+1}^{s+N} \operatorname{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right) \\ &\leq \frac{2}{\gamma N^2} \cdot \sum_{l=s+1}^{s+N} \Big( \operatorname{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l^{\text{bias}} \otimes \boldsymbol{\eta}_l^{\text{bias}}\right]\right) + \operatorname{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l^{\text{variance}} \otimes \boldsymbol{\eta}_l^{\text{variance}}\right]\right) \Big),\end{aligned} \tag{A.13}$$

where the final line follows from equation (A.9). We will now bound each of these terms to precisely characterize the excess risk of mini-batch tail-averaged SGD. We refer to the bias error of the tail-averaged iterate as the following:

$$\mathbb{E}\left[L(\overline{\mathbf{w}}^{\text{bias}}_{s+1,N})\right] - L(\mathbf{w}^*) \overset{\text{def}}{=} \frac{2}{\gamma N^2} \sum_{l=s+1}^{s+N} \operatorname{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l^{\text{bias}} \otimes \boldsymbol{\eta}_l^{\text{bias}}\right]\right) \tag{A.14}$$

Similarly, we refer to the variance error of the tail-averaged iterate as the following:

$$\mathbb{E}\left[L(\overline{\mathbf{w}}^{\text{variance}}_{s+1,N})\right] - L(\mathbf{w}^*) \overset{\text{def}}{=} \frac{2}{\gamma N^2} \sum_{l=s+1}^{s+N} \operatorname{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l^{\text{variance}} \otimes \boldsymbol{\eta}_l^{\text{variance}}\right]\right) \tag{A.15}$$

## *A.3 Lemmas for bounding the bias error*

**Lemma 22.** *With $\gamma \le \frac{\gamma_{b,max}}{2} = \frac{b}{R^2 \cdot \rho_m + (b-1)\|\mathbf{H}\|_2}$, the following bound holds:*

$$\left\| \mathbb{E}\left[(\mathbf{I} - \frac{\gamma}{b}\sum_{j=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li})(\mathbf{I} - \frac{\gamma}{b}\sum_{j=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li})\right] \right\|_2 \le 1 - \gamma\mu.$$

*Proof.* This lemma generalizes one appearing in [53] to the mini-batch size $b$ case. Denote by $\mathbf{U}$ the matrix of interest and consider the following:

$$\begin{aligned}
\mathbf{U} &= \mathbb{E}\left[(\mathbf{I} - \frac{\gamma}{b}\sum_{j=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li})(\mathbf{I} - \frac{\gamma}{b}\sum_{j=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li})\right] \\
&= \mathbf{I} - \gamma\mathbf{H} - \gamma\mathbf{H} + \left(\frac{\gamma}{b}\right)^2 \cdot \left(b\mathbb{E}\left[\|\mathbf{x}\|^2 \mathbf{x}\mathbf{x}^\top\right] + b(b-1)\mathbf{H}^2\right) \\
&\preceq \mathbf{I} - 2\gamma\mathbf{H} + \frac{\gamma^2}{b} \cdot \left(R^2\mathbf{H} + (b-1)\|\mathbf{H}\|_2\right)\mathbf{H} \\
&= \mathbf{I} - \gamma\mathbf{H},
\end{aligned}$$

from which a spectral norm bound implied by the lemma naturally follows. □

**Lemma 23.** *For any learning rate $\gamma \le \gamma_{b,max}/2$, the bias error of the tail-averaged iterate $\overline{\mathbf{w}}^{bias}_{s+1,N}$ is upper bounded as:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}}^{bias}_{s+1,N})\right] - L(\mathbf{w}^*) \le \frac{2}{\gamma^2 N^2 \mu^2}(1-\gamma\mu)^{s+1} \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right).$$

*Proof.* Before writing out the proof of the bound in the lemma, we require to bound the per step contraction properties of an SGD update in the case of the bias error (i.e. $\boldsymbol{\zeta}_{\cdot} = 0$):

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{\eta}_l\|^2\right] &= \mathbb{E}\left[\boldsymbol{\eta}_{l-1}^\top(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{li}\otimes\mathbf{x}_{li})(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{li}\otimes\mathbf{x}_{li})\boldsymbol{\eta}_{l-1}\right] \\
&= \mathbb{E}\left[\boldsymbol{\eta}_{l-1}^\top\mathbb{E}\left[(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{li}\otimes\mathbf{x}_{li})(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{li}\otimes\mathbf{x}_{li})\Big|\mathcal{F}_{l-1}\right]\boldsymbol{\eta}_{l-1}\right] \\
&\le (1-\gamma\mu)\mathbb{E}\left[\|\boldsymbol{\eta}_{l-1}\|^2\right] \quad \text{(using Lemma 22).}
\end{aligned}$$

This implies that a recursive application of the above bound yields $\mathbb{E}\left[\|\boldsymbol{\eta}_l\|^2\right] \le (1-\gamma\mu)^l\mathbb{E}\left[\|\boldsymbol{\eta}_0\|^2\right]$.

Next, we consider the bias error from equation (A.14):

$$\begin{aligned}
\mathbb{E}\left[L(\overline{\mathbf{w}}^{\text{bias}}_{s+1,N})\right] - L(\mathbf{w}^*) &= \frac{2}{\gamma N^2}\sum_{t=s+1}^{s+N}\mathbb{E}\left[\|\boldsymbol{\eta}_t\|^2\right] \\
&\le \frac{2}{\gamma N^2}\sum_{t=s+1}^{\infty}\mathbb{E}\left[\|\boldsymbol{\eta}_t\|^2\right] \\
&\le \frac{2}{\gamma N^2}\sum_{t=s+1}^{\infty}(1-\gamma\mu)^t\|\boldsymbol{\eta}_0\|^2 \\
&= \frac{2}{\gamma N^2}(\gamma\mu)^{-1}(1-\gamma\mu)^{s+1}\|\boldsymbol{\eta}_0\|^2 \\
&= \frac{2}{\gamma^2\mu N^2}(1-\gamma\mu)^{s+1}\|\boldsymbol{\eta}_0\|^2 \\
&= \frac{2}{\gamma^2\mu^2 N^2}(1-\gamma\mu)^{s+1}\cdot\Big(\mu\cdot\|\boldsymbol{\eta}_0\|^2\Big) \\
&\le \frac{2}{\gamma^2\mu^2 N^2}(1-\gamma\mu)^{s+1}\cdot\Big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\Big),
\end{aligned}$$

where in the final line, we use the fact that $\mu\mathbf{I} \preceq \mathbf{H}$. This proves the claimed bound. □

**Lemma 24.** *For any learning rate $\gamma \le \gamma_{b,max}/2$, the bias error of the* **final** *iterate $\mathbf{w}_N^{bias}$ is upper bounded as:*

$$\mathbb{E}\left[L(\mathbf{w}_N^{bias})\right] - L(\mathbf{w}^*) \le \frac{\kappa}{2}\cdot(1-\gamma\mu)^N\cdot\big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\big).$$

*Proof.* Similar to the tail-averaged case, we require to bound the per step contraction properties of an SGD update in the case of the bias error (i.e. $\boldsymbol{\zeta}_{\cdot} = 0$):

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{\eta}_N\|^2\right] &= \mathbb{E}\left[\boldsymbol{\eta}_{N-1}^\top(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{Ni}\otimes\mathbf{x}_{Ni})(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{Ni}\otimes\mathbf{x}_{Ni})\boldsymbol{\eta}_{N-1}\right] \\
&= \mathbb{E}\left[\boldsymbol{\eta}_{N-1}^\top\mathbb{E}\left[(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{Ni}\otimes\mathbf{x}_{Ni})(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{Ni}\otimes\mathbf{x}_{Ni})\middle|\mathcal{F}_{N-1}\right]\boldsymbol{\eta}_{N-1}\right] \\
&\le (1-\gamma\mu)\mathbb{E}\left[\|\boldsymbol{\eta}_{N-1}\|^2\right] \quad \text{(using Lemma 22).}
\end{aligned}$$

This implies that a recursive application of the above bound yields $\mathbb{E}\left[\|\boldsymbol{\eta}_N\|^2\right] \le (1-\gamma\mu)^N\mathbb{E}\left[\|\boldsymbol{\eta}_0\|^2\right]$. Then,

$$\begin{aligned}
\mathbb{E}\left[L(\mathbf{w}_N^{\text{bias}})\right] - L(\mathbf{w}^*) &= \frac{1}{2}\operatorname{Tr}\left((\boldsymbol{\eta}_N^{\text{bias}})^\top\mathbf{H}\boldsymbol{\eta}_N^{\text{bias}}\right) \\
&\le \frac{\lambda_{\max}(\mathbf{H})}{2}\operatorname{Tr}\left(\|\boldsymbol{\eta}_N^{\text{bias}}\|^2\right) \\
&\le \frac{\lambda_{\max}(\mathbf{H})(1-\gamma\mu)^N}{2\lambda_{\min}(\mathbf{H})}\operatorname{Tr}\left(\lambda_{\min}(\mathbf{H})\|\boldsymbol{\eta}_0\|^2\right) \\
&\le \frac{\lambda_{\max}(\mathbf{H})(1-\gamma\mu)^N}{2\lambda_{\min}(\mathbf{H})}\Big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\Big) \qquad \text{(since, } \mathbf{w}_0 = \mathbf{w}_0^{\text{bias}}\text{).} \\
&\le \frac{\kappa}{2}\cdot(1-\gamma\mu)^N\Big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\Big).
\end{aligned}$$

□

## *A.4 Lemmas for bounding the variance error*

Now, we seek to understand the behavior of the variance error of the tail-averaged iterate $\overline{\mathbf{w}}_{s+1,N}$. We begin by noting here that the variance error is analyzed by beginning the optimization at the solution, i.e. $\boldsymbol{\eta}_0^{\text{variance}} = 0$ and allowing the noise to drive the process. In particular, we write out the recursive updates that characterize the variance error:

$$\boldsymbol{\eta}_t^{\text{variance}} = \mathbf{P}_{tb}\boldsymbol{\eta}_{t-1}^{\text{variance}} + \gamma\boldsymbol{\zeta}_{tb}, \text{ with } \boldsymbol{\eta}_0^{\text{variance}} = 0.$$

This implies that by defining $\boldsymbol{\Phi}_t^{\text{variance}} \overset{\text{def}}{=} \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{variance}}\otimes\boldsymbol{\eta}_t^{\text{variance}}\right]$, we have:

$$\boldsymbol{\Phi}_t^{\text{variance}} = \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{variance}}\otimes\boldsymbol{\eta}_t^{\text{variance}}\right]$$

$$\begin{aligned}
&= \mathbb{E}\left[\mathbb{E}\left[\left(\mathbf{P}_{tb}\boldsymbol{\eta}_{t-1}^{\text{variance}} + \gamma\boldsymbol{\zeta}_{tb}\right) \otimes \left(\mathbf{P}_{tb}\boldsymbol{\eta}_{t-1}^{\text{variance}} + \gamma\boldsymbol{\zeta}_{tb}\right)|\mathcal{F}_{t-1}\right]\right] \\
&= \mathbb{E}\left[\mathbf{P}_{tb}\boldsymbol{\Phi}_{t-1}^{\text{variance}}\mathbf{P}_{tb}^{\top}\right] + \frac{\gamma^2}{b}\boldsymbol{\Sigma}. \qquad \text{(A.16)}
\end{aligned}$$

where, $\mathcal{F}_{t-1}$ is the filtration defined using the samples $\{\mathbf{x}_{ji}, y_{ji}\}_{j=1,i=1}^{j=t-1,i=b}$. Furthermore cross terms are zero since $\mathbb{E}\left[\boldsymbol{\zeta}_{tb}|\mathcal{F}_{t-1}\right] = 0$ owing to first order optimality conditions. Recounting that $\mathbf{P}_{tb} = \mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\otimes\mathbf{x}_{ti}$, we express equation (A.16) using a linear operator as follows:

$$\begin{aligned}
\mathbb{E}\left[\mathbf{P}_{tb}\boldsymbol{\Phi}_{t-1}^{\text{variance}}\mathbf{P}_{tb}^{\top}\right] &= \mathbb{E}\left[\left(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\right)\boldsymbol{\Phi}_{t-1}^{\text{variance}}\left(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\right)\right] \\
&\overset{\text{def}}{=} (\mathcal{I} - \gamma\mathcal{T}_b)\boldsymbol{\Phi}_{t-1}^{\text{variance}},
\end{aligned}$$

with $\mathcal{T}_b$ representing the following linear operator:

$$\mathcal{T}_b = \mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}} - \frac{\gamma}{b}\mathcal{M} - \gamma\frac{b-1}{b}\mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}},$$

with $\mathcal{M} = \mathbb{E}\left[\mathbf{x}\otimes\mathbf{x}\otimes\mathbf{x}\otimes\mathbf{x}\right]$, $\mathcal{H}_{\mathcal{L}} = \mathbf{H}\otimes\mathbf{I}$ and $\mathcal{H}_{\mathcal{R}} = \mathbf{I}\otimes\mathbf{H}$ representing the left and right multiplication linear operators corresponding to the matrix $\mathbf{H}$. Given this notation, we consider $\boldsymbol{\Phi}_t^{\text{variance}}$:

$$\begin{aligned}
\boldsymbol{\Phi}_t^{\text{variance}} &= (\mathcal{I} - \gamma\mathcal{T}_b)\boldsymbol{\Phi}_{t-1}^{\text{variance}} + \frac{\gamma^2}{b}\boldsymbol{\Sigma} \\
&= \frac{\gamma^2}{b}\left(\sum_{k=0}^{t-1}(\mathcal{I} - \gamma\mathcal{T}_b)^k\right)\boldsymbol{\Sigma}. \qquad \text{(A.17)}
\end{aligned}$$

Before bounding the variance error, we will describe a lemma that shows that the expected covariance of the variance error $\boldsymbol{\Phi}_t^{\text{variance}}$ initialized at 0 grows monotonically to its steady state value (in a PSD sense).

**Lemma 25.** *The sequence of centered variance iterates $\boldsymbol{\eta}_t^{variance}$ have expected covariances that monotonically grow in a PSD sense, i.e.:*

$$0 = \boldsymbol{\Phi}_0^{variance} \preceq \boldsymbol{\Phi}_1^{variance} \preceq \boldsymbol{\Phi}_2^{variance} .... \preceq \boldsymbol{\Phi}_\infty^{variance}.$$

*Proof.* This lemma generalizes the lemma appearing in [53, 54]. We begin by recounting the $t^{\text{th}}$ variance iterate, i.e.:

$$\boldsymbol{\eta}_t^{\text{variance}} = \gamma \sum_{j=1}^{t} \mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b}.$$

This implies in particular that

$$\begin{aligned}
\boldsymbol{\Phi}_t^{\text{variance}} &= \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{variance}} \otimes \boldsymbol{\eta}_t^{\text{variance}}\right] \\
&= \gamma^2 \sum_{j=1}^{t}\sum_{l=1}^{t} \mathbb{E}\left[\mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b} \otimes \boldsymbol{\zeta}_{l,b}\mathbf{Q}_{l+1,b}^{\top}\right] \quad \text{(from equation (A.4))} \\
&= \gamma^2 \sum_{j=1}^{t}\sum_{l=1}^{t} \mathbb{E}\left[\mathbf{Q}_{j+1,t}\mathbb{E}\left[\boldsymbol{\zeta}_{j,b} \otimes \boldsymbol{\zeta}_{l,b}|\mathcal{F}_{j-1}\right]\mathbf{Q}_{l+1,b}^{\top}\right] \\
&= \gamma^2 \sum_{j=1}^{t} \mathbb{E}\left[\mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b} \otimes \boldsymbol{\zeta}_{j,b}\mathbf{Q}_{j+1,t}^{\top}\right] \\
&= \frac{\gamma^2}{b} \sum_{j=1}^{t} \mathbb{E}\left[\mathbf{Q}_{j+1,t}\boldsymbol{\Sigma}\mathbf{Q}_{j+1,t}^{\top}\right].
\end{aligned}$$

where, the third line follows since $\mathbb{E}\left[\boldsymbol{\zeta}_{l,b} \otimes \boldsymbol{\zeta}_{j,b}\right] = 0$ for $j \neq l$, similar to arguments in equation A.3. This immediately reveals that the sequence of covariances grows as a function of time, since,

$$\boldsymbol{\Phi}_{t+1}^{\text{variance}} - \boldsymbol{\Phi}_t^{\text{variance}} = \frac{\gamma^2}{b}\mathbb{E}\left[\mathbf{Q}_{2,t+1}\boldsymbol{\Sigma}\mathbf{Q}_{2,t+1}^{\top}\right] \succeq 0.$$

□

This lemma leads to a natural upper bound on the variance error, as expressed below:

**Lemma 26.** *With $\gamma < \frac{\gamma_{b,max}}{2}$, the variance error of the tail-averaged iterate $\overline{\mathbf{w}}_{s+1,N}^{variance}$ is upper bounded as:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{variance})\right] - L(\mathbf{w}^*) \leq \frac{2}{Nb}\operatorname{Tr}\left(\mathcal{T}_b^{-1}\boldsymbol{\Sigma}\right).$$

*Proof.* Considering the variance error of tail-averaged iterate from equation (A.15):

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{\text{variance}})\right] - L(\mathbf{w}^*) = \frac{2}{\gamma N^2} \cdot \sum_{l=s+1}^{s+N}\Bigg(\operatorname{Tr}\left(\mathbb{E}\left[\boldsymbol{\Phi}_l^{\text{variance}}\right]\right)\Bigg)$$

$$\begin{aligned}
&\leq \frac{2}{\gamma N} \cdot \mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{\Phi}_\infty^{\text{variance}}\right]\right) \qquad \text{(from Lemma 25)} \\
&= \frac{2}{\gamma N} \cdot \frac{\gamma^2}{b} \cdot \mathrm{Tr}\left(\sum_{k=0}^{\infty} (\mathcal{I} - \gamma \mathcal{T}_b)^k \boldsymbol{\Sigma}\right) \qquad \text{(from equation (A.4))} \\
&= \frac{2}{Nb} \mathrm{Tr}\,(\mathcal{T}_b^{-1} \boldsymbol{\Sigma}).
\end{aligned}$$

□

**Lemma 27.** *With $\gamma < \frac{\gamma_{b,max}}{2}$, the variance error of the* **final** *iterate $\mathbf{w}_N^{variance}$, obtained by running mini-batch SGD for N steps is upper bounded as:*

$$\mathbb{E}\left[L(\mathbf{w}_N^{variance})\right] - L(\mathbf{w}^*) \leq \frac{\gamma}{2b} \mathrm{Tr}\left(\mathbf{H} \mathcal{T}_b^{-1} \boldsymbol{\Sigma}\right).$$

*Proof.* We note that since we deal with the square loss case,

$$\begin{aligned}
\mathbb{E}\left[L(\mathbf{w}_N^{\text{variance}})\right] - L(\mathbf{w}^*) &= \frac{1}{2} \mathrm{Tr}\,(\mathbf{H} \boldsymbol{\Phi}_N^{\text{variance}}) \\
&\leq \frac{1}{2} \mathrm{Tr}\,(\mathbf{H} \boldsymbol{\Phi}_\infty^{\text{variance}}) \quad \text{(using Lemma 25)} \\
&= \frac{\gamma^2}{2b} \mathrm{Tr}\left(\mathbf{H} \sum_{j=0}^{\infty} (\mathcal{I} - \gamma \mathcal{T}_b)^j \boldsymbol{\Sigma}\right) \\
&= \frac{\gamma}{2b} \mathrm{Tr}\left(\mathbf{H} \mathcal{T}_b^{-1} \boldsymbol{\Sigma}\right)
\end{aligned}$$

□

**Lemma 28.** *Denoting the assumption (A) $\gamma \leq \gamma_{b,max}/2$,*

1. *With (A) in place, $\mathcal{T}_b \succeq 0$.*

2. *With (A) in place, $\mathcal{T}_b^{-1} \mathbf{W} \succeq 0$ for every $\mathbf{W} \in \mathcal{S}(d),\ \mathbf{W} \succeq 0$*

3. $\mathrm{Tr}\left((\mathcal{H}_\mathcal{R} + \mathcal{H}_\mathcal{L})^{-1} \mathbf{A}\right) = \frac{1}{2} \mathrm{Tr}\,(\mathbf{H}^{-1} \mathbf{A})\ \ \forall\ \mathbf{A} \in \mathcal{S}^+(\mathbb{R}^d)$

4. *With (A) in place,*

$$\mathrm{Tr}\left(\mathcal{T}_b^{-1} \boldsymbol{\Sigma}\right) \leq 2\, \mathrm{Tr}\left(\mathbf{H}^{-1} \boldsymbol{\Sigma}\right)$$

*Proof. Proof of claim 1 in Lemma 28*: $\mathcal{T}_b \succeq 0$ implies that for all symmetric matrices $\mathbf{A} \in \mathcal{S}(d)$, we have $\operatorname{Tr}(\mathbf{A}\mathcal{T}_b\mathbf{A}) \geq 0$, and this is true owing to the following inequalities:

$$\begin{aligned}
\langle \mathbf{A}, \mathcal{T}_b\mathbf{A}\rangle &= 2\operatorname{Tr}(\mathbf{AHA}) - \frac{\gamma}{b}\mathbb{E}\left[\langle \mathbf{x}, \mathbf{Ax}\rangle^2\right] - \frac{\gamma(b-1)}{b}\langle \mathbf{H}, \mathbf{AHA}\rangle \\
&\geq 2\operatorname{Tr}(\mathbf{AHA}) - \frac{\gamma}{b}\mathbb{E}\left[\|\mathbf{x}\|^2 \|\mathbf{Ax}\|^2\right] - \frac{\gamma(b-1)}{b}\|\mathbf{H}\|\operatorname{Tr}(\mathbf{AHA}) \\
&\geq 2\operatorname{Tr}(\mathbf{AHA}) - \frac{\gamma}{b}R^2\mathbb{E}\left[\|\mathbf{Ax}\|^2\right] - \frac{\gamma(b-1)}{b}\|\mathbf{H}\|\operatorname{Tr}(\mathbf{AHA}) \\
&\geq \Big(2 - \frac{\gamma}{b}\left(R^2 + (b-1)\|\mathbf{H}\|\right)\Big)\operatorname{Tr}(\mathbf{AHA}),
\end{aligned}$$

and using the definition of $\gamma_{b,\max}$ finishes the claim.

*Proof of claim 2 in Lemma 28*: We require to prove $\mathcal{T}_b{}^{-1}$ operating on a PSD matrix produces a PSD matrix, or in other words, $\mathcal{T}_b{}^{-1}$ is a PSD map.

$$\begin{aligned}
\mathcal{T}_b{}^{-1} &= [\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} - \frac{\gamma}{b}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})]^{-1} \\
&= (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{\frac{1}{2}}[\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} - \frac{\gamma}{b}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})]^{-1} \\
&\qquad\qquad (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{\frac{1}{2}}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}} \\
&= (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}[\mathcal{I} - \frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}]^{-1} \\
&\qquad\qquad (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}} \qquad \text{(A.18)}
\end{aligned}$$

Now, we prove that $\|\frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}\| < 1$. Given $\gamma < \gamma_{b,\max}/2$, we employ claim 1 to note that $\mathcal{T}_b \succ 0$.

$$\begin{aligned}
&\mathcal{T}_b \succ 0 \\
&\quad \Rightarrow \mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} - \frac{\gamma}{b}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}) \succ 0 \\
&\quad \Rightarrow \frac{\gamma}{b}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}) \prec \mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} \\
&\quad \Rightarrow \frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}} \prec \mathcal{I} \\
&\quad \Rightarrow \|\frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}\| < 1
\end{aligned}$$

With this fact in place, we employ Taylor series to expand $\mathcal{T}_b{}^{-1}$ in equation (A.18), i.e.:

$$\mathcal{T}_b{}^{-1} = (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}\sum_{i=0}^{\infty}(\frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}})^i(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}$$

$$= \sum_{i=0}^{\infty} (\frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-1}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}))^i (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-1}$$

The proof completes by employing the following facts: Using Lyapunov's Theorem ([10] Proposition A 1.2.6), we know $(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-1}$ is a PSD map, i.e. if $(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-1}(A) = B$, then, if $A$ is PSD $\implies$ $B$ is PSD. Furthermore, $\mathcal{M}$ is also a PSD map, i.e. if $A_1$ is PSD, $\mathcal{M}(A_1) = E[(x^T A_1 x) x \otimes x]$ is PSD as well. Finally, $\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}$ is also a PSD map, since, if $A_2$ is PSD, then, $\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}(A_2) = H A_2 H$ which is PSD as well. With all these facts in place, we note that each term in the Taylor's expansion above is a PSD map implying the overall map is PSD as well, thus rounding up the proof to claim 2 in Lemma 28.

*Proof of claim 3 in Lemma 28*:

We know that the operator $(\mathcal{H}_\mathcal{R} + \mathcal{H}_\mathcal{L})^{-1}$ is a PSD map, i.e, it maps PSD matrices to PSD matrices. Since $\mathbf{A} \succeq 0$, we replace this condition with $\mathbf{U} = (\mathcal{H}_\mathcal{R} + \mathcal{H}_\mathcal{L})^{-1}\mathbf{A} \succeq 0$ implying, we need to show the following:

$$\mathrm{Tr}\left(\mathbf{U}\right) = \frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\mathbf{A}\right) \ \forall \ \mathbf{U} \succeq 0$$

Examining the right hand side, we see the following:

$$\begin{aligned} \frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\mathbf{A}\right) &= \frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})\mathbf{U}\right) \\ &= \frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\mathbf{H}\mathbf{U} + \mathbf{H}^{-1}\mathbf{U}\mathbf{H}\right) \\ &= \mathrm{Tr}\left(\mathbf{U}\right) \end{aligned}$$

thus wrapping up the proof of claim 4.

*Proof of claim 4 in Lemma 28*: Let $\mathcal{U} = \mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} - \frac{\gamma}{b}\cdot(b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}$. Then,

$$\begin{aligned} \mathcal{T}_b^{-1}\mathbf{\Sigma} &= \left(\mathcal{U} - \frac{\gamma}{b}\mathcal{M}\right)^{-1}\mathbf{\Sigma} \\ &= \sum_{i=0}^{\infty}\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)^i \mathcal{U}^{-1}\mathbf{\Sigma}. \end{aligned}$$

Let $\mathbf{A} = \mathcal{U}^{-1}\mathbf{\Sigma}$, $\mathbf{A}' = \mathcal{U}^{-1}\mathbf{H}$. Then,

$$\mathcal{T}_b^{-1}\mathbf{\Sigma} = \sum_{i=1}^{\infty}\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)^i \mathbf{A}.$$

The $i = 0$ term is just $= \mathbf{A}$. Now, considering $i = 1$, we have:

$$\begin{aligned}\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\mathbf{A} &\preceq \frac{\gamma}{b}\|\mathbf{A}\|_2\mathcal{U}^{-1}\mathcal{M}\mathbf{I}\\ &\preceq \frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\mathcal{U}^{-1}\mathbf{H} = \frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\mathbf{A}'.\end{aligned}$$

Next, considering $i = 2$, we have:

$$\begin{aligned}\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)^2\mathbf{A} &= \left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)\cdot\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)\mathbf{A}\\ &\preceq \left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right)\cdot\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)\mathbf{A}'\\ &\preceq \left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right)\cdot\left(\frac{\gamma}{b}\mathcal{U}^{-1}\right)\cdot\|\mathbf{A}'\|_2\cdot R^2\mathbf{H}\\ &\preceq \left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right)\cdot\left(\frac{\gamma}{b}\|\mathbf{A}'\|_2 R^2\right)\cdot\mathbf{A}'.\end{aligned}$$

Noting this recursive structure, we see that:

$$\begin{aligned}\mathcal{T}_b^{-1}\mathbf{\Sigma} &= \sum_{i=0}^{\infty}\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)^i\mathbf{A}\\ &\preceq \mathbf{A} + \sum_{i=1}^{\infty}\left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right)\cdot\left(\frac{\gamma}{b}\|\mathbf{A}'\|_2 R^2\right)^{i-1}\cdot\mathbf{A}'\\ &= \mathbf{A} + \frac{\left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right)}{1-\left(\frac{\gamma}{b}\|\mathbf{A}'\|_2 R^2\right)}\cdot\mathbf{A}'.\end{aligned}$$

Note that this summation is finite iff $\gamma \le \frac{b}{R^2\|\mathbf{A}'\|_2}$. Further, applying the trace operator on both sides, we have:

$$\operatorname{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) \le \operatorname{Tr}(\mathbf{A}) + \frac{\left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right)}{1-\left(\frac{\gamma}{b}\|\mathbf{A}'\|_2 R^2\right)}\operatorname{Tr}(\mathbf{A}'). \tag{A.19}$$

Now, for any psd matrix $\mathbf{Z} \succeq 0$, let us upperbound $\mathcal{U}^{-1}\mathbf{Z}$:

$$\mathcal{U}^{-1}\mathbf{Z} = \sum_{j=0}^{\infty}\left(\gamma\cdot\frac{b-1}{b}\cdot(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\cdot\mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}}\right)^i(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}.$$

The recursion can be bounded by analyzing $i = 1$:

$$\begin{aligned}
&\gamma \cdot \frac{b-1}{b} \cdot (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1} \cdot \mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}} \cdot (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{Z} \\
&\preceq \|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{Z}\|_2 \cdot \gamma \cdot \frac{b-1}{b} \cdot (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1} \cdot \mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}} \cdot \mathbf{I} \\
&\preceq \|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{Z}\|_2 \cdot \gamma \cdot \frac{b-1}{b} \cdot (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1} \cdot \|\mathbf{H}\|_2\mathbf{H} \\
&= \|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{Z}\|_2 \cdot \gamma \cdot \frac{b-1}{2b} \cdot \|\mathbf{H}\|_2 \cdot \mathbf{I}
\end{aligned}$$

This indicates the means to recurse for bounding terms $i \geq 2$:

$$\begin{aligned}
\mathcal{U}^{-1}\mathbf{Z} &\preceq \sum_{j=0}^{\infty} \|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{Z}\|_2 \left(\gamma \cdot \frac{b-1}{2b} \cdot \|\mathbf{H}\|_2\right)^j \cdot \mathbf{I} \\
&= \frac{\|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{Z}\|_2}{1 - \gamma \cdot \frac{(b-1)\|\mathbf{H}\|_2}{2b}} \cdot \mathbf{I}.
\end{aligned}$$

The upperbound above is true as long as $\gamma < \frac{2b}{(b-1)\|\mathbf{H}\|_2}$. This now allows us to obtain bounds on $\|\mathbf{A}\|_2, \|\mathbf{A}'\|_2, \mathrm{Tr}\,(\mathbf{A}')$:

$$\begin{aligned}
\|\mathbf{A}\|_2 &\leq \frac{\|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\|_2}{1 - \gamma \cdot \frac{b-1}{2b} \cdot \|\mathbf{H}\|_2} \\
\|\mathbf{A}'\|_2 &\leq \frac{1/2}{1 - \gamma \cdot \frac{b-1}{2b} \cdot \|\mathbf{H}\|_2} \\
\mathrm{Tr}\,(\mathbf{A}') &\leq \frac{d/2}{1 - \gamma \cdot \frac{b-1}{2b} \cdot \|\mathbf{H}\|_2}.
\end{aligned}$$

Substituting these in equation (A.19):

$$\mathrm{Tr}\left(\mathcal{T}_b{}^{-1}\mathbf{\Sigma}\right) \leq \mathrm{Tr}\,(\mathbf{A}) + \frac{\frac{\gamma R^2}{2b} \cdot d\|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\|_2}{\left(1 - \frac{\gamma}{2b} \cdot (R^2 + (b-1)\|\mathbf{H}\|_2)\right) \cdot \left(1 - \gamma \cdot \frac{b-1}{2b}\|\mathbf{H}\|_2\right)}. \tag{A.20}$$

with the conditions on $\gamma$ being: $\gamma \leq \frac{2b}{(b-1)\|\mathbf{H}\|_2}$, $\gamma \leq \frac{2b}{R^2+(b-1)\|\mathbf{H}\|_2}$, $\gamma \leq \frac{2b}{R^2}$. These are combined using $\gamma \leq \frac{2b}{R^2+(b-1)\|\mathbf{H}\|_2}$. Once this condition is satisfied, the denominator of the second term can be upperbounded by atmost a constant. Next, looking at the numerator of the second term, we see that $\gamma \leq \frac{2b}{R^2 \cdot \frac{d\|(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\|_2}{\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right)}} = \frac{2b}{R^2\rho_{\mathrm{m}}}$ allows for the second term to be upperbounded by $\mathcal{O}(\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right))$. This is clearly satisfied if $\gamma \leq \frac{2b}{R^2\cdot\rho_{\mathrm{m}}+(b-1)\|\mathbf{H}\|_2}$. In particular, setting $\gamma$ to be half of this maximum, we have:

$$\mathrm{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) \leq \mathrm{Tr}\left(\mathbf{A}\right) + 2\,\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right). \tag{A.21}$$

Next, for obtaining a sharp bound on $\mathrm{Tr}\left(\mathbf{A}\right)$, we require comparing $\mathrm{Tr}\left(\hat{\mathbf{\Sigma}}\right) = \mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}}-\gamma\cdot\frac{b-1}{b}\cdot\mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right)$ with $\mathrm{Tr}\left(\tilde{\mathbf{\Sigma}}\right) = \mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right)$. For this, without loss of generality, we can consider $\mathbf{H}$ to be diagonal, and this implies that comparing the diagonal elements of $\hat{\mathbf{\Sigma}}_{ii} = \mathbf{\Sigma}_{ii}/(2\lambda_i - \gamma\frac{b-1}{b}\lambda_i^2)$ while $\tilde{\mathbf{\Sigma}}_{ii} = \mathbf{\Sigma}_{ii}/2\lambda_i$. Comparing these, we see that

$$\begin{aligned}\mathrm{Tr}\left(\hat{\mathbf{\Sigma}}\right) = \mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}}-\gamma\cdot\frac{b-1}{b}\cdot\mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right) &\leq \frac{1}{1-\gamma\frac{b-1}{2b}\|\mathbf{H}\|_2}\,\mathrm{Tr}\left(\tilde{\mathbf{\Sigma}}\right)\\ &= \frac{1}{1-\gamma\frac{b-1}{2b}\|\mathbf{H}\|_2}\,\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right).\end{aligned}$$

Noting that $\mathrm{Tr}\left(\mathbf{A}\right) = \mathrm{Tr}\left(\hat{\mathbf{\Sigma}}\right)$, we see that substituting the above in equation (A.21), we have:

$$\begin{aligned}\mathrm{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) &\leq \frac{1}{1-\gamma\frac{b-1}{2b}\|\mathbf{H}\|_2}\,\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right) + 2\,\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right)\\ &\leq 4\,\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right) = 2\,\mathrm{Tr}\left(\mathbf{H}^{-1}\mathbf{\Sigma}\right).\end{aligned}$$

□

**Corollary 29.** *Consider the agnostic case of the streaming LSR problem. With* $\gamma \leq \frac{\gamma_{b,max}}{2}$*, the variance error of the tail-averaged iterate* $\overline{\mathbf{w}}_{s+1,N}^{variance}$ *is upper bounded as:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{variance})\right] - L(\mathbf{w}^*) \leq \frac{4}{Nb}\cdot\widehat{\sigma_{MLE}^2}.$$

*Proof.* The result follows in a straightforward manner by noting that $\gamma \leq \frac{\gamma_{b,\max}}{2}$ implying that $\mathrm{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) \leq 2\,\mathrm{Tr}\left(\mathbf{H}^{-1}\mathbf{\Sigma}\right)$ and by substituting into the result of Lemma 26. □

**Corollary 30.** *With* $\gamma \leq \frac{\gamma_{b,max}}{2}$, $\mathbf{\Sigma} = \sigma^2\mathbf{H}$ *the variance error of the* **final** *iterate* $\mathbf{w}_N^{variance}$*, obtained by running mini-batch SGD for N steps is upper bounded as:*

$$\mathbb{E}\left[L(\mathbf{w}_N^{variance})\right] - L(\mathbf{w}^*) \leq \frac{\gamma\sigma^2}{2b}\,\mathrm{Tr}\,\mathbf{H}.$$

*Proof.* This follows from the fact that $\mathcal{T}_b^{-1}\mathbf{\Sigma} \preceq \sigma^2\mathbf{I}$, implying that $\mathbf{H}\mathcal{T}_b^{-1}\mathbf{\Sigma} \preceq \sigma^2\mathbf{H}$ and then applying the trace operator on the result of Lemma 27. □

## A.5 Main Results

### A.5.1 Derivation of Divergent Learning Rate

A necessary condition for the convergence of Stochastic Gradient Updates is $\mathcal{T}_b \succeq 0$, and this by definition implies,

$$\begin{aligned}
&\langle \mathbf{W}, \mathcal{T}_b\mathbf{W}\rangle \geq 0, \quad \mathbf{W} \in \mathcal{S}(d) \\
\implies & 2\,\mathrm{Tr}\,(\mathbf{WHW}) - \frac{\gamma}{b}\,\mathrm{Tr}\,(\mathbf{W}\mathcal{M}\mathbf{W}) - \gamma(\frac{b-1}{b})\,\mathrm{Tr}\,(\mathbf{WHWH}) \geq 0 \\
\implies & \frac{2}{\gamma} \geq \frac{\mathrm{Tr}\,(\mathbf{W}\mathcal{M}\mathbf{W}) + (b-1)\,\mathrm{Tr}\,(\mathbf{WHWH})}{b\,\mathrm{Tr}\,(\mathbf{WHW})} \\
\implies & \frac{2}{\gamma^{div}_{b,\max}} = \sup_{\mathbf{W}\in\mathcal{S}(d)} \frac{\mathrm{Tr}\,(\mathbf{W}\mathcal{M}\mathbf{W}) + (b-1)\,\mathrm{Tr}\,(\mathbf{WHWH})}{b\,\mathrm{Tr}\,(\mathbf{WHW})}.
\end{aligned}$$

### A.5.2 Proof of Theorem 1

*Proof of Theorem 1.* The proof of Theorem 1 follows from characterizing bias-variance decomposition for the tail-averaged iterate in section A.2.3 with equation (A.13).

The bias error of the tail-averaged iterate (equation (A.14)) is bounded with Lemma 22 and Lemma 23 in section A.3.

The variance error of the tail-averaged iterate (equation (A.15)) is bounded with Lemma 25, Lemma 26, Lemma 28 and Corollary 29 in Section A.4.

The final expression follows through substituting the result of Lemma 23 and Corollary 29 into equation (A.13), with appropriate parameters of the problem, i.e., with a batch size $b$, number of burn-in iterations $s$, number of tail-averaged iterations $n/b - s$ to provide the claimed excess risk bound of Algorithm 1: The final expression follows through substituting the result of Lemma 23 and Corollary 29 into equation (A.13), with appropriate parameters of the problem, i.e., with a batch size $b$, number of burn-in iterations $s$, number of tail-averaged iterations $n/b - s$ to provide the claimed excess risk bound of Algorithm 1:

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \frac{2}{\gamma^2\mu^2(\frac{n}{b}-s)^2} \cdot (1-\gamma\mu)^s \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + 4 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}}}{b \cdot (\frac{n}{b}-s)}.$$

□

## A.5.3 Proof of Lemma 4

*Proof of Lemma 4.* The proof of Lemma 4 follows from characterizing bias-variance decomposition for the final iterate in section A.2.2 with equation (A.8).

The bias error of the final iterate is bounded with Lemma 22 and Lemma 24 in Section A.3.

The variance error of the final iterate is bounded with Lemma 25, Lemma 27, Lemma 28 and Corollary 30 in Section A.4.

The final expression follows through substituting the result of Lemma 24 and Corollary 30 into equation A.8, with appropriate parameters of the problem, i.e., with a batch size $b$, number of samples $n$ and number of iterations $\lfloor n/b \rfloor$, to provide the claimed excess risk bound:

$$\mathbb{E}\left[L(\mathbf{w}_{\lfloor n/b \rfloor})\right] - L(\mathbf{w}^*) \leq \kappa_b(1-\gamma\mu)^{\lfloor n/b \rfloor}\left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + \frac{\gamma}{b}\sigma^2 \operatorname{Tr}(\mathbf{H}),$$

□

## A.5.4 Proof of Theorem 5

*Proof.* Let $\widetilde{L_e} = \mathbb{E}\left[L(\mathbf{w}_e)\right] - L(\mathbf{w}^*)$. We will first provide a recursive bound for $\widetilde{L_e}$ for $e \leq \log\left(\frac{n}{bt}\right) - 1$ using Theorem 1, with a mini-batch size of $b_e = 1 + 2^{e-1}b$, where, $b = b_{\text{thresh}} - 1$, $n_e = b_e \cdot t$, $s = t-1$:

$$\begin{aligned}\widetilde{L_e} &\leq 2\kappa_{b_e}^2 \exp\left(-\frac{n_e}{b_e \cdot \kappa_{b_e}}\right)\widetilde{L_{e-1}} + 4\frac{\widehat{\sigma^2_{\text{MLE}}}}{b_e} \\ &\leq \exp\left(-\frac{n_e}{3b_e\kappa_e\log(\kappa_e)}\right) \cdot \widetilde{L_{e-1}} + 4 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}}}{b_e}.\end{aligned}$$

Next, denote $\kappa = \|\mathbf{H}\|_2/\mu$; now, let us bound $\kappa_{b_e}$:

$$\begin{aligned}
\kappa_{b_e} &= \frac{R^2 \cdot \frac{d\left\|(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right\|_2}{\mathrm{Tr}((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma})} + (b_e - 1)\|\mathbf{H}\|_2}{b_e \mu} \\
&= \kappa \cdot \frac{b_{\text{thresh}} - 1 + b_e - 1}{b_e} = \kappa \cdot \frac{b_{\text{thresh}} - 1 + 2^{e-1}(b_{\text{thresh}} - 1)}{2^{e-1}(b_{\text{thresh}} - 1)} \\
&= \kappa \cdot \frac{1 + 2^{e-1}}{2^{e-1}} \leq 2\kappa.
\end{aligned}$$

This implies $\kappa_{b_e}\log(\kappa_{b_e}) \leq 4\kappa\log(\kappa)$. This implies, revisiting the recursion on $\widetilde{L_e}$, we have:

$$\begin{aligned}
\widetilde{L_e} &\leq \exp\left( - \frac{n_e}{12 b_e \kappa \log(\kappa)}\right) \cdot \widetilde{L_{e-1}} + 4 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}}}{b_e} \\
&\leq \exp\left( - \frac{t}{12\kappa\log(\kappa)}\right) \cdot \widetilde{L_{e-1}} + 4 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}}}{2^{e-1}b} \\
&\leq \exp\left( - \frac{te}{12\kappa\log(\kappa)}\right) \cdot \widetilde{L_0} + \frac{4\widehat{\sigma^2_{\text{MLE}}}}{b} \cdot \sum_{j=1}^{e} \frac{\exp\left( - \frac{t(j-1)}{12\kappa\log(\kappa)}\right)}{2^{e-j}} \\
&\leq \exp\left( - \frac{te}{12\kappa\log(\kappa)}\right) \cdot \widetilde{L_0} + \frac{4\widehat{\sigma^2_{\text{MLE}}}}{b} \cdot \frac{1/2^{e-1}}{1 - 2\cdot\exp\left( - \frac{t}{12\kappa\log\kappa}\right)} \\
&\leq \exp\left( - \frac{te}{12\kappa\log(\kappa)}\right) \cdot \widetilde{L_0} + \frac{12\widehat{\sigma^2_{\text{MLE}}}}{2^e b} \quad (\text{since } t > 24\kappa\log(\kappa)) \\
&= \exp\left( - \frac{te}{12\kappa\log(\kappa)}\right) \cdot \widetilde{L_0} + \frac{12\widehat{\sigma^2_{\text{MLE}}}}{b\cdot n} \cdot (4bt) \quad (\text{since } 2^e = n/(4bt)) \\
&= \exp\left( - \frac{te}{12\kappa\log(\kappa)}\right) \cdot \widetilde{L_0} + 48 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}} t}{n} \qquad \text{(A.22)}
\end{aligned}$$

Next, for the final epoch, we have $b = n/2t$, $s = t/2$, and a total of $n/2$ samples, implying:

$$\begin{aligned}
\widetilde{L_{e+1}} &\leq \frac{2\kappa_b^2}{\left(t/2\right)^2} \cdot \exp\left( - \frac{t}{2\kappa_b}\right)\widetilde{L_e} + 4 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}}}{b\cdot\left(n/4b\right)} = \frac{8\kappa_b^2}{t^2} \cdot \exp\left( - \frac{t}{2\kappa_b}\right) \cdot \widetilde{L_e} + 16 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}}}{n} \\
&\leq \frac{32\kappa^2}{t^2} \cdot \exp\left( - \frac{t}{4\kappa}\right) \cdot \widetilde{L_e} + 16 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}}}{n} \quad (\text{since } \kappa_b \leq 2\kappa) \\
&\leq \frac{32\kappa^2}{t^2} \cdot \exp\left( - \frac{t}{4\kappa}\right) \cdot \left( \exp\left( - \frac{te}{12\kappa\log(\kappa)}\right) \cdot \widetilde{L_0} + 48 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}} t}{n}\right) + 16 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}}}{n}
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{32\kappa^2}{t^2}\cdot\exp\big(-\frac{t}{4\kappa}\big)\cdot\exp\left(-\frac{te}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_0}+64\kappa\exp\big(-t/4\kappa\big)\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}+16\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&\leq \frac{32\kappa^2}{t^2}\cdot\exp\big(-\frac{t}{4\kappa}\big)\cdot\exp\left(-\frac{te}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_0}+80\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&\leq \exp\left(-\frac{t(e+1)}{12\kappa\log(\kappa)}\right)\cdot\widetilde{L_0}+80\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&=\left(\frac{2bt}{n}\right)^{\frac{t}{12\kappa\log(\kappa)}}\widetilde{L_0}+80\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}, \qquad\qquad (A.23)
\end{aligned}
$$

which rounds up the proof of the theorem. □

### *A.5.5 Proof of Theorem 6*

*Proof.* For analyzing the parameter mixing scheme, we require tracking the progress of the $i^{th}$ machine's SGD updates using its centered estimate $\boldsymbol{\eta}_k^{(i)}$. Furthermore, the tail-averaged iterate for the $i^{th}$ machine is representeed as $\bar{\boldsymbol{\eta}}^{(i)} \overset{\text{def}}{=} \frac{1}{N}\sum_{k=s+1}^{s+N}\boldsymbol{\eta}_k^{(i)}$. Finally, the model averaged estimate is represented with its own centered estimate defined as $\bar{\boldsymbol{\eta}} = \frac{1}{P}\sum_{i=1}^{P}\bar{\boldsymbol{\eta}}^{(i)}$. Now, in a manner similar to standard mini-batch tail-averaged SGD on a single machine, the model averaged iterate admits its own bias variance decomposition, through which $\bar{\boldsymbol{\eta}} = \bar{\boldsymbol{\eta}}^{\text{bias}} + \bar{\boldsymbol{\eta}}^{\text{variance}}$ and an upperbound on the excess risk is written as:

$$
\begin{aligned}
\mathbb{E}\left[L(\overline{\mathbf{w}})\right]-L(\mathbf{w}^*) &= \mathbb{E}\left[\frac{1}{2}\langle(\overline{\mathbf{w}}-\mathbf{w}^*),\mathbf{H}(\overline{\mathbf{w}}-\mathbf{w}^*)\rangle\right]=\mathbb{E}\left[\frac{1}{2}\langle\bar{\boldsymbol{\eta}},\mathbf{H}\bar{\boldsymbol{\eta}}\rangle\right]\\
&\leq \mathbb{E}\left[\langle\bar{\boldsymbol{\eta}}^{\text{bias}},\mathbf{H}\bar{\boldsymbol{\eta}}^{\text{bias}}\rangle\right]+\mathbb{E}\left[\langle\bar{\boldsymbol{\eta}}^{\text{variance}},\mathbf{H}\bar{\boldsymbol{\eta}}^{\text{variance}}\rangle\right].
\end{aligned}
$$

We will first handle the variance since it is straightforward given that the noise $\boldsymbol{\zeta}$ is independent for different machines SGD runs. What this implies is the following:

$$
\begin{aligned}
\bar{\boldsymbol{\eta}}^{\text{variance}} &= \frac{1}{P}\sum_{i=1}^{P}\bar{\boldsymbol{\eta}}^{(i),\text{variance}}\\
\implies \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{variance}}\otimes\bar{\boldsymbol{\eta}}^{\text{variance}}\right] &= \frac{1}{P^2}\sum_{i,j}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{variance}}\otimes\bar{\boldsymbol{\eta}}^{(j),\text{variance}}\right]\\
&= \frac{1}{P^2}\Bigg(\sum_{i}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{variance}}\otimes\bar{\boldsymbol{\eta}}^{(i),\text{variance}}\right]
\end{aligned}
$$

$$+ \sum_{i \neq j} \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{variance}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{variance}}\right] \Bigg)$$

$$= \frac{1}{P}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(1),\text{variance}} \otimes \bar{\boldsymbol{\eta}}^{(1),\text{variance}}\right]. \tag{A.24}$$

Where, the final line follows because $\forall\ i \neq j$, the terms are in expectation equal to zero since in expectation each of the noise terms is zero (from first order optimality conditions). The other observation is that the only terms left are $P$ independent runs of tail-averaged SGD in each of the machine, whose risk is straightforward to bound from Corollary 29. This implies

$$\langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{variance}} \otimes \bar{\boldsymbol{\eta}}^{\text{variance}}\right]\rangle \leq \frac{4}{PNb} \cdot \widehat{\sigma^2_{\text{MLE}}}. \quad \text{(using Corollary 29)} \tag{A.25}$$

Next, let us consider the bias error:

$$\begin{aligned}
\bar{\boldsymbol{\eta}}^{\text{bias}} &= \frac{1}{P}\sum_{i=1}^{P} \bar{\boldsymbol{\eta}}^{(i),\text{bias}} \\
\implies \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{\text{bias}}\right] &= \frac{1}{P^2}\sum_{i,j} \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right] \\
&= \frac{1}{P^2}\Bigg( \sum_{i=1}^{P} \underbrace{\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(i),\text{bias}}\right]}_{\text{independent runs of tail-averaged SGD}} + \sum_{i \neq j} \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right] \Bigg),
\end{aligned} \tag{A.26}$$

which implies that we require bounding $\forall\ i \neq j$, $\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right]$.

$$\begin{aligned}
\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right] &= \frac{1}{N^2}\sum_{k,l=s+1}^{s+N} \mathbb{E}\left[\boldsymbol{\eta}_k^{(i),\text{bias}} \otimes \boldsymbol{\eta}_l^{(j),\text{bias}}\right] = \frac{1}{N^2}\sum_{k,l=s+1}^{s+N} \mathbb{E}\left[\boldsymbol{\eta}_k^{(i),\text{bias}}\right] \otimes \mathbb{E}\left[\boldsymbol{\eta}_l^{(j),\text{bias}}\right] \\
&= \frac{1}{N^2}\sum_{k,l=s+1}^{s+N} \mathbb{E}\left[\mathbf{Q}_{1:k}^{(i)}\boldsymbol{\eta}_0\right] \otimes \mathbb{E}\left[\mathbf{Q}_{1:l}^{(j)}\boldsymbol{\eta}_0\right] \quad \text{(from equation (A.5))} \\
&= \frac{1}{N^2}\Bigg(\sum_{k=s+1}^{s+N} (\mathbf{I} - \gamma\mathbf{H})^k\Bigg)\boldsymbol{\eta}_0 \otimes \boldsymbol{\eta}_0\Bigg(\sum_{l=s+1}^{s+N} (\mathbf{I} - \gamma\mathbf{H})^l\Bigg) \\
&\preceq \frac{1}{N^2}\Bigg(\sum_{k=s+1}^{\infty} (\mathbf{I} - \gamma\mathbf{H})^k\Bigg)\boldsymbol{\eta}_0 \otimes \boldsymbol{\eta}_0\Bigg(\sum_{l=s+1}^{\infty} (\mathbf{I} - \gamma\mathbf{H})^l\Bigg) \\
&= \frac{1}{\gamma^2 N^2}\mathbf{H}^{-1}(\mathbf{I} - \gamma\mathbf{H})^{s+1}\boldsymbol{\eta}_0 \otimes \boldsymbol{\eta}_0(\mathbf{I} - \gamma\mathbf{H})^{s+1}\mathbf{H}^{-1}.
\end{aligned}$$

This implies that,

$$\begin{aligned}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}}\otimes\bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right] &\leq \frac{1}{\gamma^2N^2}\cdot\langle\mathbf{H},\mathbf{H}^{-1}(\mathbf{I}-\gamma\mathbf{H})^{s+1}\boldsymbol{\eta}_0\otimes\boldsymbol{\eta}_0(\mathbf{I}-\gamma\mathbf{H})^{s+1}\mathbf{H}^{-1}\rangle\\ &= \frac{1}{\gamma^2N^2}\cdot\boldsymbol{\eta}_0^\top(\mathbf{I}-\gamma\mathbf{H})^{s+1}\mathbf{H}^{-1}\mathbf{H}\mathbf{H}^{-1}(\mathbf{I}-\gamma\mathbf{H})^{s+1}\boldsymbol{\eta}_0\\ &\leq \frac{(1-\gamma\mu)^{2s+2}}{\mu\gamma^2N^2}\|\boldsymbol{\eta}_0\|^2 \leq \frac{(1-\gamma\mu)^{2s+2}}{\mu^2\gamma^2N^2}\cdot\Big(L(\mathbf{w}_0)-L(\mathbf{w}^*)\Big). \qquad \text{(A.27)}\end{aligned}$$

Combining the bound for the cross terms in equation (A.27) and Lemma 23 for the self-terms, we get:

$$\langle\mathbf{H},\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{bias}}\otimes\bar{\boldsymbol{\eta}}^{\text{bias}}\right]\rangle \leq \frac{(1-\gamma\mu)^{s+1}}{\mu^2\gamma^2N^2}\cdot\frac{2+(1-\gamma\mu)^{s+1}\cdot(P-1)}{P}\cdot\Big(L(\mathbf{w}_0)-L(\mathbf{w}^*)\Big). \qquad \text{(A.28)}$$

The proof wraps up by substituting the relation $N = n/(P\cdot b) - s$ in equations (A.25) and (A.28). □

### *A.5.6 Proof of Lemma 3*

For this problem instance, we begin by noting that $(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}$ is diagonal as well, with entries:

$$\{(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}\}_{ii} = \frac{1}{2}\{\mathbf{H}^{-1}\boldsymbol{\Sigma}\}_{ii} = \begin{cases}1/2 & \text{if } i=1\\ 1/2(d-1) & \text{if } i>1\end{cases}$$

Let us consider the case with batch size $b=1$. With the appropriate choice of step size $\gamma$ that ensure contracting operators, we require considering $\operatorname{Tr}\left(\mathcal{T}_b^{-1}\boldsymbol{\Sigma}\right)$ as in equation A.19, which corresponds to bounding the leading order term in the variance. We employ the taylor's expansion (just as in Claim 2 of Lemma 28) to expand the term of interest $\mathcal{T}_b^{-1}\boldsymbol{\Sigma}$:

$$\begin{aligned}\mathcal{T}_b^{-1}\boldsymbol{\Sigma} &= \sum_{i=0}^{\infty}\left(\gamma(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathcal{M}\right)^i(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}\\ &= (\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}+\sum_{i=1}^{\infty}\left(\gamma(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathcal{M}\right)^i(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}\\ \Rightarrow \operatorname{Tr}\mathcal{T}_b^{-1}\boldsymbol{\Sigma} &= \operatorname{Tr}(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}+\sum_{i=1}^{\infty}\operatorname{Tr}\left[\left(\gamma(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathcal{M}\right)^i(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}\right]\end{aligned}$$

$$\operatorname{Tr} \mathcal{T}_b^{-1}\mathbf{\Sigma} = \frac{1}{2}\operatorname{Tr}\mathbf{H}^{-1}\mathbf{\Sigma} + \sum_{i=1}^{\infty}\operatorname{Tr}\left[\left(\gamma(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathcal{M}\right)^{i}(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right]$$

We observe that the term corresponding to $i = 0$ works out regardless of the choice of stepsize $\gamma$; we then switch our attention to the second term, i.e., the term corresponding to $i = 1$:

$$\operatorname{Tr}\left(\gamma(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathcal{M}\right)(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma} = \frac{d+2}{4}\cdot\operatorname{Tr}(\mathbf{\Sigma})$$

We require that this term should be $\leq \operatorname{Tr}(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}$, implying,

$$\gamma < \frac{4\operatorname{Tr}(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}}{(d+2)\operatorname{Tr}(\mathbf{\Sigma})}$$

For this example, we observe that this yields $\gamma < \frac{4}{(d+2)(1+\frac{1}{d})}$, which clearly is off by a factor $d$ compared to the well-specified case which requires $\gamma < \frac{d}{(d+2)(1+\frac{1}{d})}$, thus indicating a clear separation between the step sizes required by SGD for the well-specified and mis-specified cases.

### *A.5.7 Proofs of supporting lemmas*

**Proof of Lemma 7**

*Proof of Lemma 7.* We begin by considering $\langle \mathbf{I}, \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{bias}}\otimes\boldsymbol{\eta}_t^{\text{bias}}\right]\rangle$:

$$\begin{aligned}\langle \mathbf{I}, \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{bias}}\otimes\boldsymbol{\eta}_t^{\text{bias}}\right]\rangle &= \mathbb{E}\left[\|\boldsymbol{\eta}_t^{\text{bias}}\|^2\right]\\ &= \mathbb{E}\left[(\boldsymbol{\eta}_{t-1}^{\text{bias}})^\top\left(\mathbf{I}-\frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\mathbf{x}_{ti}^\top\right)\left(\mathbf{I}-\frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\mathbf{x}_{ti}^\top\right)\boldsymbol{\eta}_{t-1}^{\text{bias}}\right]\\ &\leq (1-\gamma\mu)\cdot\mathbb{E}\left[\|\boldsymbol{\eta}_{t-1}^{\text{bias}}\|^2\right] \qquad \text{(from Lemma 22)},\end{aligned}$$

from where the lemma follows through substitution of $\gamma = \gamma_{b,\max}/2$. □

**Proof of Lemma 8**

*Proof of Lemma 8.* From equation A.17, we have that:

$$\mathbf{\Phi}_t^{\text{variance}} = \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{variance}}\otimes\boldsymbol{\eta}_t^{\text{variance}}\right]$$

$$= \frac{\gamma^2}{b}\bigg(\sum_{k=0}^{t-1}(\mathcal{I} - \gamma\mathcal{T}_b)^k\bigg)\mathbf{\Sigma}$$

Allowing $t \to \infty$, we have:

$$\mathbf{\Phi}_{\infty}^{\text{variance}} = \frac{\gamma}{b}\mathcal{T}_b^{-1}\mathbf{\Sigma} \preceq \frac{\gamma}{b}\cdot\sigma^2\mathbf{I} \quad \text{(from claim 4 in lemma 28 since } \gamma \leq \gamma_{b,\max}/2,\ \mathbf{\Sigma} = \sigma^2\mathbf{H}\text{).}$$

Substituting $\gamma = \gamma_{b,\max}/2$, the result follows. $\square$

# Appendix B

# APPENDIX: ACCELERATING STOCHASTIC GRADIENT DESCENT (FOR LEAST SQUARES REGRESSION)

## B.1 Appendix setup

We will first provide a note on the organization of the appendix and follow that up with introducing the notations.

### B.1.1 Organization

- In subsection B.1.2, we will recall notation from the main chapter and introduce some new notation that will be used across the appendix.

- In section B.2, we will write out expressions that characterize the generalization error of the proposed accelerated SGD method. In order to bound the generalization error, we require developing an understanding of two terms namely the bias error and the variance error.

- In section B.3, we prove lemmas that will be used in subsequent sections to prove bounds on the bias and variance error.

- In section B.4, we will bound the bias error of the proposed accelerated stochastic gradient method. In particular, Lemma 12 is the key lemma that provides a new potential function with which this chapter achieves acceleration. Further, Lemma 40 is the lemma that bounds all the terms of the bias error.

- In section B.5, we will bound the variance error of the proposed accelerated stochastic gradient method. In particular, Lemma 14 is the key lemma that considers a stochastic process view of the proposed accelerated stochastic gradient method and provides a sharp bound on the covariance of the stationary distribution of the iterates. Furthermore, Lemma 44 bounds all terms of the variance error.

- Section B.6 presents the proof of Theorem 9. In particular, this section aggregates the result of Lemma 40 (which bounds all terms of the bias error) and Lemma 44 (which bounds all terms of the variance error) to present the guarantees of Algorithm 3.

### *B.1.2 Notations*

We begin by introducing $\mathcal{M}$, which is the fourth moment tensor of the input $\mathbf{a} \sim \mathcal{D}$, i.e.:

$$\mathcal{M} \overset{\text{def}}{=} \mathbb{E}_{(\mathbf{a},b)\sim\mathcal{D}}\left[\mathbf{a}\otimes\mathbf{a}\otimes\mathbf{a}\otimes\mathbf{a}\right]$$

Applying the fourth moment tensor $\mathcal{M}$ to any matrix $\mathbf{S}\in\mathbb{R}^{d\times d}$ produces another matrix in $\mathbb{R}^{d\times d}$ that is expressed as:

$$\mathcal{M}\mathbf{S} \overset{\text{def}}{=} \mathbb{E}\left[(\mathbf{a}^\top\mathbf{S}\mathbf{a})\mathbf{a}\mathbf{a}^\top\right].$$

With this definition in place, we recall $R^2$ as the smallest number, such that $\mathcal{M}$ applied to the identity matrix $\mathbf{I}$ satisfies:

$$\mathcal{M}\mathbf{I} = \mathbb{E}\left[\|\mathbf{a}\|_2^2\,\mathbf{a}\mathbf{a}^\top\right] \preceq R^2\ \mathbf{H}$$

Moreover, we recall that the condition number of the distribution $\kappa = R^2/\mu$, where $\mu$ is the smallest eigenvalue of $\mathbf{H}$. Furthermore, the definition of the statistical condition number $\widetilde{\kappa}$ of the distribution follows by applying the fourth moment tensor $\mathcal{M}$ to $\mathbf{H}^{-1}$, i.e.:

$$\mathcal{M}\mathbf{H}^{-1} = \mathbb{E}\left[(\mathbf{a}^\top\mathbf{H}^{-1}\mathbf{a})\cdot\mathbf{a}\mathbf{a}^\top\right] \preceq \widetilde{\kappa}\ \mathbf{H}$$

We denote by $\mathcal{A}_{\mathcal{L}}$ and $\mathcal{A}_{\mathcal{R}}$ the left and right multiplication operator of any matrix $\mathbf{A}\in\mathbb{R}^{d\times d}$, i.e. for any matrix $\mathbf{S}\in\mathbb{R}^{d\times d}$, $\mathcal{A}_{\mathcal{L}}\mathbf{S} = \mathbf{A}\mathbf{S}$ and $\mathcal{A}_{\mathcal{R}}\mathbf{S} = \mathbf{S}\mathbf{A}$.

**Parameter choices:** In all of appendix we choose the parameters in Algorithm 3 as

$$\alpha = \frac{\sqrt{\kappa\widetilde{\kappa}}}{c_2\sqrt{2c_1 - c_1^2} + \sqrt{\kappa\widetilde{\kappa}}}, \quad \beta = c_3 \frac{c_2\sqrt{2c_1 - c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}, \quad \gamma = c_2 \frac{\sqrt{2c_1 - c_1^2}}{\mu\sqrt{\kappa\widetilde{\kappa}}}, \quad \delta = \frac{c_1}{R^2}$$

where $c_1$ is an arbitrary constant satisfying $0 < c_1 < \frac{1}{2}$. Furthermore, we note that $c_3 = \frac{c_2\sqrt{2c_1 - c_1^2}}{c_1}$, $c_2^2 = \frac{c_4}{2-c_1}$ and $c_4 < 1/6$. Note that we recover Theorem 9 by choosing $c_1 = 1/5, c_2 = \sqrt{5}/9, c_3 = \sqrt{5}/3, c_4 = 1/9$. We denote

$$c \overset{\text{def}}{=} \alpha(1 - \beta) \text{ and, } q \overset{\text{def}}{=} \alpha\delta + (1 - \alpha)\gamma.$$

Recall that $\mathbf{x}^*$ denotes unique minimizer of $P(\mathbf{x})$, i.e. $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^d} \mathbb{E}_{(\mathbf{a},b)\sim\mathcal{D}}\left[(b - \langle \mathbf{x}, \mathbf{a}\rangle)^2\right]$. We track $\boldsymbol{\theta}_k = \begin{bmatrix} \mathbf{x}_k - \mathbf{x}^* \\ \mathbf{y}_k - \mathbf{x}^* \end{bmatrix}$. The following equation captures the updates of Algorithm 3:

$$\begin{aligned} \boldsymbol{\theta}_{k+1} &= \begin{bmatrix} 0 & \mathbf{I} - \delta\widehat{\mathbf{H}}_{k+1} \\ -c\cdot\mathbf{I} & (1+c)\cdot\mathbf{I} - q\cdot\widehat{\mathbf{H}}_{k+1} \end{bmatrix} \boldsymbol{\theta}_k + \begin{bmatrix} \delta\cdot\epsilon_{k+1}\mathbf{a}_{k+1} \\ q\cdot\epsilon_{k+1}\mathbf{a}_{k+1} \end{bmatrix} \\ &\overset{\text{def}}{=} \widehat{\mathbf{A}}_{k+1}\boldsymbol{\theta}_k + \boldsymbol{\zeta}_{k+1}, \end{aligned} \tag{B.1}$$

where, $\widehat{\mathbf{H}}_{k+1} \overset{\text{def}}{=} \mathbf{a}_{k+1}\mathbf{a}_{k+1}^\top$, $\widehat{\mathbf{A}}_{k+1} \overset{\text{def}}{=} \begin{bmatrix} 0 & \mathbf{I} - \delta\widehat{\mathbf{H}}_{k+1} \\ -c\cdot\mathbf{I} & (1+c)\cdot\mathbf{I} - q\cdot\widehat{\mathbf{H}}_{k+1} \end{bmatrix}$ and $\boldsymbol{\zeta}_{k+1} \overset{\text{def}}{=} \begin{bmatrix} \delta\cdot\epsilon_{k+1}\mathbf{a}_{k+1} \\ q\cdot\epsilon_{k+1}\mathbf{a}_{k+1} \end{bmatrix}$.

Furthermore, we denote by $\boldsymbol{\Phi}_k$ the expected covariance of $\boldsymbol{\theta}_k$, i.e.:

$$\boldsymbol{\Phi}_k \overset{\text{def}}{=} \mathbb{E}\left[\boldsymbol{\theta}_k \otimes \boldsymbol{\theta}_k\right].$$

Next, let $\mathcal{F}_k$ denote the filtration generated by samples $\{(\mathbf{a}_1, b_1), \cdots, (\mathbf{a}_k, b_k)\}$. Then,

$$\mathbf{A} \overset{\text{def}}{=} \mathbb{E}\left[\widehat{\mathbf{A}}_{k+1}|\mathcal{F}_k\right] = \begin{bmatrix} 0 & \mathbf{I} - \delta\mathbf{H} \\ -c\mathbf{I} & (1+c)\mathbf{I} - q\mathbf{H} \end{bmatrix}.$$

By iterated conditioning, we also have

$$\mathbb{E}\left[\boldsymbol{\theta}_{k+1}|\mathcal{F}_k\right] = \mathbf{A}\boldsymbol{\theta}_k. \tag{B.2}$$

Without loss of generality, we assume that $\mathbf{H}$ is a diagonal matrix. We now note that we can rearrange the coordinates through an eigenvalue decomposition so that $\mathbf{A}$ becomes a block-diagonal matrix with $2 \times 2$ blocks. We denote the $j^{\text{th}}$ block by $\mathbf{A}_j$:

$$\mathbf{A}_j \overset{\text{def}}{=} \begin{bmatrix} 0 & 1 - \delta\lambda_j \\ -c & 1 + c - q\lambda_j \end{bmatrix},$$

where $\lambda_j$ denotes the $j^{\text{th}}$ eigenvalue of $\mathbf{H}$. Next,

$$\begin{aligned} \mathcal{B} &\overset{\text{def}}{=} \mathbb{E}\left[\widehat{\mathbf{A}}_{k+1} \otimes \widehat{\mathbf{A}}_{k+1} | \mathcal{F}_k\right], \text{ and} \\ \widehat{\mathbf{\Sigma}} &\overset{\text{def}}{=} \mathbb{E}\left[\boldsymbol{\zeta}_{k+1} \otimes \boldsymbol{\zeta}_{k+1} | \mathcal{F}_k\right] = \begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{\Sigma} \preceq \sigma^2 \cdot \begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{H}. \end{aligned}$$

Finally, we observe the following:

$$\begin{aligned} \mathbb{E}\left[(\mathbf{A} - \widehat{\mathbf{A}}_{k+1}) \otimes (\mathbf{A} - \widehat{\mathbf{A}}_{k+1}) | \mathcal{F}_k\right] &= \mathbf{A} \otimes \mathbf{A} - \mathbb{E}\left[\widehat{\mathbf{A}}_{k+1} \otimes \mathbf{A} | \mathcal{F}_k\right] \\ &\quad - \mathbb{E}\left[\widehat{\mathbf{A}}_{k+1} \otimes \mathbf{A} | \mathcal{F}_k\right] + \mathbb{E}\left[\widehat{\mathbf{A}}_{k+1} \otimes \widehat{\mathbf{A}}_{k+1} | \mathcal{F}_k\right] \\ &= -\mathbf{A} \otimes \mathbf{A} + \mathbb{E}\left[\widehat{\mathbf{A}}_{k+1} \otimes \widehat{\mathbf{A}}_{k+1} | \mathcal{F}_k\right] \\ \implies \mathbb{E}\left[\widehat{\mathbf{A}}_{k+1} \otimes \widehat{\mathbf{A}}_{k+1} | \mathcal{F}_k\right] &= \mathbb{E}\left[(\mathbf{A} - \widehat{\mathbf{A}}_{k+1}) \otimes (\mathbf{A} - \widehat{\mathbf{A}}_{k+1}) | \mathcal{F}_k\right] + \mathbf{A} \otimes \mathbf{A} \end{aligned}$$

We now define:

$$\begin{aligned} \mathcal{R} &\overset{\text{def}}{=} \mathbb{E}\left[(\mathbf{A} - \widehat{\mathbf{A}}_{k+1}) \otimes (\mathbf{A} - \widehat{\mathbf{A}}_{k+1}) | \mathcal{F}_k\right], \text{ and} \\ \mathcal{D} &\overset{\text{def}}{=} \mathbf{A} \otimes \mathbf{A}. \end{aligned}$$

Thus implying the following relation between the operators $\mathcal{B}, \mathcal{D}$ and $\mathcal{R}$:

$$\mathcal{B} = \mathcal{D} + \mathcal{R}.$$

## *B.2 The Tail-Average Iterate: Covariance and bias-variance decomposition*

We begin by considering the first-order Markovian recursion as defined by equation (B.1):

$$\boldsymbol{\theta}_j = \widehat{\mathbf{A}}_j \boldsymbol{\theta}_{j-1} + \boldsymbol{\zeta}_j.$$

We refer by $\mathbf{\Phi}_j$ the covariance of the $j^{\text{th}}$ iterate, i.e.:

$$\mathbf{\Phi}_j \overset{\text{def}}{=} \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right] \tag{B.3}$$

Consider a decomposition of $\boldsymbol{\theta}_j$ as $\boldsymbol{\theta}_j = \boldsymbol{\theta}_j^{\text{bias}} + \boldsymbol{\theta}_j^{\text{variance}}$, where $\boldsymbol{\theta}_j^{\text{bias}}$ and $\boldsymbol{\theta}_j^{\text{variance}}$ are defined as follows:

$$\boldsymbol{\theta}_j^{\text{bias}} \overset{\text{def}}{=} \widehat{\mathbf{A}}_j \boldsymbol{\theta}_{j-1}^{\text{bias}}; \qquad \boldsymbol{\theta}_0^{\text{bias}} \overset{\text{def}}{=} \boldsymbol{\theta}_0, \text{ and} \tag{B.4}$$

$$\boldsymbol{\theta}_j^{\text{variance}} \overset{\text{def}}{=} \widehat{\mathbf{A}}_j \boldsymbol{\theta}_{j-1}^{\text{variance}} + \boldsymbol{\zeta}_j; \qquad \boldsymbol{\theta}_0^{\text{variance}} \overset{\text{def}}{=} 0. \tag{B.5}$$

We note that

$$\mathbb{E}\left[\boldsymbol{\theta}_j^{\text{bias}}\right] = \mathbf{A}\mathbb{E}\left[\boldsymbol{\theta}_{j-1}^{\text{bias}}\right], \tag{B.6}$$

$$\mathbb{E}\left[\boldsymbol{\theta}_j^{\text{variance}}\right] = \mathbf{A}\mathbb{E}\left[\boldsymbol{\theta}_{j-1}^{\text{variance}}\right]. \tag{B.7}$$

Note equation (B.7) follows using a conditional expectation argument with the fact that $\mathbb{E}\left[\boldsymbol{\zeta}_k\right] = 0 \ \forall\ k$ owing to first order optimality conditions.

Before we prove the decomposition holds using an inductive argument, let us understand what the bias and variance sub-problem intuitively mean.

Note that the *bias* sub-problem (defined by equation (B.4)) refers to running algorithm on the noiseless problem (i.e., where, $\boldsymbol{\zeta}_{\boldsymbol{\cdot}} = 0$ a.s.) by starting it at $\boldsymbol{\theta}_0^{\text{bias}} = \boldsymbol{\theta}_0$. The bias essentially measures the dependence of the generalization error on the excess risk of the initial point $\boldsymbol{\theta}_0$ and bears similarities to convergence rates studied in the context of offline optimization.

The *variance* sub-problem (defined by equation (B.5)) measures the dependence of the generalization error on the noise introduced during the course of optimization, and this is associated with the statistical aspects of the optimization problem. The variance can be understood as starting the algorithm at the solution ($\boldsymbol{\theta}_0^{\text{variance}} = 0$) and running the optimization driven solely by noise. Note that the variance is associated with sharp statistical lower bounds which dictate its rate of decay as a function of the number of oracle calls $n$.

Now, we will prove that the decomposition $\boldsymbol{\theta}_j = \boldsymbol{\theta}_j^{\text{bias}} + \boldsymbol{\theta}_j^{\text{variance}}$ captures the recursion expressed in equation (B.1) through induction. For the base case $j = 1$, we see that

$$\begin{aligned}\boldsymbol{\theta}_1 &= \widehat{\mathbf{A}}_1\boldsymbol{\theta}_0 + \boldsymbol{\zeta}_1\\ &= \underbrace{\widehat{\mathbf{A}}_1\boldsymbol{\theta}_0^{\text{bias}}}_{\because\ \boldsymbol{\theta}_0^{\text{bias}}=\boldsymbol{\theta}_0} + \underbrace{\widehat{\mathbf{A}}_1\boldsymbol{\theta}_0^{\text{variance}}}_{=0,\ \because\ \boldsymbol{\theta}_0^{\text{variance}}=0} + \boldsymbol{\zeta}_1\\ &= \boldsymbol{\theta}_1^{\text{bias}} + \boldsymbol{\theta}_1^{\text{variance}}\end{aligned}$$

Now, for the inductive step, let us assume that the decomposition holds in the $j-1^{st}$ iteration, i.e. we assume $\boldsymbol{\theta}_{j-1} = \boldsymbol{\theta}_{j-1}^{\text{bias}} + \boldsymbol{\theta}_{j-1}^{\text{variance}}$. We will then prove that this relation holds in the $j^{th}$ iteration. Towards this, we will write the recursion:

$$\begin{aligned}\boldsymbol{\theta}_j &= \widehat{\mathbf{A}}_j\boldsymbol{\theta}_{j-1} + \boldsymbol{\zeta}_j\\ &= \widehat{\mathbf{A}}_j(\boldsymbol{\theta}_{j-1}^{\text{bias}} + \boldsymbol{\theta}_{j-1}^{\text{variance}}) + \boldsymbol{\zeta}_j \quad \text{(using the inductive hypothesis)}\\ &= \widehat{\mathbf{A}}_j\boldsymbol{\theta}_{j-1}^{\text{bias}} + \widehat{\mathbf{A}}_j\boldsymbol{\theta}_{j-1}^{\text{variance}} + \boldsymbol{\zeta}_j\\ &= \boldsymbol{\theta}_j^{\text{bias}} + \boldsymbol{\theta}_j^{\text{variance}}.\end{aligned}$$

This proves the decomposition holds through a straight forward inductive argument.

In a similar manner as $\boldsymbol{\theta}_j$, the tail-averaged iterate $\bar{\boldsymbol{\theta}}_{t,n} \overset{\text{def}}{=} \frac{1}{n-t}\sum_{j=t+1}^{n}\boldsymbol{\theta}_j$ can also be written as $\bar{\boldsymbol{\theta}}_{t,n} = \bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}} + \bar{\boldsymbol{\theta}}_{t,n}^{\text{variance}}$, where $\bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}} \overset{\text{def}}{=} \frac{1}{n-t}\sum_{j=t+1}^{n}\boldsymbol{\theta}_j^{\text{bias}}$ and $\bar{\boldsymbol{\theta}}_{t,n}^{\text{variance}} \overset{\text{def}}{=} \frac{1}{n-t}\sum_{j=t+1}^{n}\boldsymbol{\theta}_j^{\text{variance}}$. Furthermore, the tail-averaged iterate $\bar{\boldsymbol{\theta}}_{t,n}$ and its bias and variance counterparts $\bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}}, \bar{\boldsymbol{\theta}}_{t,n}^{\text{variance}}$ are associated with their corresponding covariance matrices $\bar{\boldsymbol{\Phi}}_{t,n}, \bar{\boldsymbol{\Phi}}_{t,n}^{\text{bias}}, \bar{\boldsymbol{\Phi}}_{t,n}^{\text{variance}}$ respectively. Note that $\bar{\boldsymbol{\Phi}}_{t,n}$ can be upper bounded using Cauchy-Shwartz inequality as:

$$\begin{aligned}\mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n}\otimes\bar{\boldsymbol{\theta}}_{t,n}\right] &\preceq 2\cdot\left(\mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}}\otimes\bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}}\right] + \mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n}^{\text{variance}}\otimes\bar{\boldsymbol{\theta}}_{t,n}^{\text{variance}}\right]\right)\\ \implies \bar{\boldsymbol{\Phi}}_{t,n} &\preceq 2\cdot(\bar{\boldsymbol{\Phi}}_{t,n}^{\text{bias}} + \bar{\boldsymbol{\Phi}}_{t,n}^{\text{variance}}). \qquad \text{(B.8)}\end{aligned}$$

The above inequality is referred to as the *bias-variance* decomposition and is well known from previous work [7, 35, 55], and we re-derive this decomposition for the sake of completeness.

We will now derive an expression for the covariance of the tail-averaged iterate and apply it to obtain the covariance of the bias ($\bar{\mathbf{\Phi}}_{t,n}^{\text{bias}}$) and variance ($\bar{\mathbf{\Phi}}_{t,n}^{\text{variance}}$) error of the tail-averaged iterate.

### *B.2.1 The tail-averaged iterate and its covariance*

We begin by writing out an expression for the tail-averaged iterate $\bar{\boldsymbol{\theta}}_{t,n}$ as:

$$\bar{\boldsymbol{\theta}}_{t,n} = \frac{1}{n-t}\sum_{j=t+1}^{n} \boldsymbol{\theta}_j$$

To get the excess risk of the tail-averaged iterate $\bar{\boldsymbol{\theta}}_{t,n}$, we track its covariance $\bar{\mathbf{\Phi}}_{t,n}$:

$$\begin{aligned}
\bar{\mathbf{\Phi}}_{t,n} &= \mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n} \otimes \bar{\boldsymbol{\theta}}_{t,n}\right] \\
&= \frac{1}{(n-t)^2} \sum_{j,l=t+1}^{n} \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_l\right] \\
&= \frac{1}{(n-t)^2} \sum_{j} \left( \sum_{l=t+1}^{j-1} \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_l\right] + \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right] + \sum_{l=j+1}^{n} \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_l\right] \right) \\
&= \frac{1}{(n-t)^2} \sum_{j} \left( \sum_{l=t+1}^{j-1} \mathbf{A}^{j-l}\mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right] + \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right] + \sum_{l=j+1}^{n} \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right](\mathbf{A}^\top)^{l-j} \right) \text{(from (B.2))} \\
&= \left( \sum_{l=t+1}^{n}\sum_{j=l+1}^{n} \mathbf{A}^{j-l}\mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right] + \sum_{j=t+1}^{n} \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right] + \sum_{j=t+1}^{n}\sum_{l=j+1}^{n} \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right](\mathbf{A}^\top)^{l-j} \right)/(n-t)^2 \\
&= \left( \sum_{j=t+1}^{n}\sum_{l=j+1}^{n} \mathbf{A}^{l-j}\mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right] + \sum_{j=t+1}^{n} \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right] + \sum_{j=t+1}^{n}\sum_{l=j+1}^{n} \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right](\mathbf{A}^\top)^{l-j} \right)/(n-t)^2 \\
&= \frac{1}{(n-t)^2}\Bigg( \sum_{j=t+1}^{n} (\mathbf{I}-\mathbf{A})^{-1}(\mathbf{A}-\mathbf{A}^{n+1-j})\mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right] + \sum_{j=t+1}^{n} \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right] \\
&\qquad\qquad + \sum_{j=t+1}^{n} \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right](\mathbf{I}-\mathbf{A}^\top)^{-1}(\mathbf{A}^\top-(\mathbf{A}^\top)^{n+1-j}) \Bigg) \\
&= \sum_{j=t+1}^{n} \left( \mathcal{I} + (\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}(\mathcal{A}_{\mathcal{L}} - \mathcal{A}_{\mathcal{L}}^{n+1-j}) + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^\top)^{-1}(\mathcal{A}_{\mathcal{R}}^\top - (\mathcal{A}_{\mathcal{R}}^\top)^{n+1-j}) \right) \frac{\mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right]}{(n-t)^2} \\
&= \frac{1}{(n-t)^2} \sum_{j=t+1}^{n} \left( \mathcal{I} + (\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}(\mathcal{A}_{\mathcal{L}} - \mathcal{A}_{\mathcal{L}}^{n+1-j}) + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^\top)^{-1}(\mathcal{A}_{\mathcal{R}}^\top - (\mathcal{A}_{\mathcal{R}}^\top)^{n+1-j}) \right) \mathbf{\Phi}_j.
\end{aligned} \tag{B.9}$$

Note that the above recursion can be applied to obtain the covariance of the tail-averaged iterate for the bias ($\bar{\mathbf{\Phi}}_{t,n}^{\text{bias}}$) and variance ($\bar{\mathbf{\Phi}}_{t,n}^{\text{variance}}$) error, since the conditional expectation arguments employed in obtaining equation (B.9) are satisfied by both the recursion used in tracking the bias error (i.e. equation (B.4)) and the variance error (i.e. equation (B.5)). This implies that,

$$\bar{\mathbf{\Phi}}_{t,n}^{\text{bias}} \overset{\text{def}}{=} \sum_{j=t+1}^{n} \Big(\mathcal{I} + (\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}(\mathcal{A}_{\mathcal{L}} - \mathcal{A}_{\mathcal{L}}^{n+1-j}) + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top} - (\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j})\Big)\frac{\mathbf{\Phi}_j^{\text{bias}}}{(n-t)^2} \tag{B.10}$$

$$\bar{\mathbf{\Phi}}_{t,n}^{\text{variance}} \overset{\text{def}}{=} \sum_{j=t+1}^{n} \Big(\mathcal{I} + (\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}(\mathcal{A}_{\mathcal{L}} - \mathcal{A}_{\mathcal{L}}^{n+1-j}) + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top} - (\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j})\Big)\frac{\mathbf{\Phi}_j^{\text{variance}}}{(n-t)^2} \tag{B.11}$$

### *B.2.2 Covariance of Bias error of the tail-averaged iterate*

*Proof of Lemma 11.* To obtain the covariance of the bias error of the tail-averaged iterate, we first need to obtain $\mathbf{\Phi}_j^{\text{bias}}$, which we will by unrolling the recursion of equation (B.4):

$$\begin{aligned}
\boldsymbol{\theta}_k^{\text{bias}} &= \widehat{\mathbf{A}}_k \boldsymbol{\theta}_{k-1}^{\text{bias}} \\
\implies \mathbf{\Phi}_k^{\text{bias}} &= \mathbb{E}\left[\boldsymbol{\theta}_k^{\text{bias}} \otimes \boldsymbol{\theta}_k^{\text{bias}}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\boldsymbol{\theta}_k^{\text{bias}} \otimes \boldsymbol{\theta}_k^{\text{bias}} | \mathcal{F}_{k-1}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\widehat{\mathbf{A}}_k \boldsymbol{\theta}_{k-1}^{\text{bias}} \otimes \boldsymbol{\theta}_{k-1}^{\text{bias}} \widehat{\mathbf{A}}_k^{\top} | \mathcal{F}_{k-1}\right]\right] \\
&= \mathcal{B}\ \mathbb{E}\left[\boldsymbol{\theta}_{k-1}^{\text{bias}} \otimes \boldsymbol{\theta}_{k-1}^{\text{bias}}\right] = \mathcal{B}\ \mathbf{\Phi}_{k-1}^{\text{bias}} \\
\implies \mathbf{\Phi}_k^{\text{bias}} &= \mathcal{B}^k\ \mathbf{\Phi}_0^{\text{bias}}
\end{aligned} \tag{B.12}$$

Next, we recount the equation for the covariance of the bias of the tail-averaged iterate from equation (B.10):

$$\bar{\mathbf{\Phi}}_{t,n}^{\text{bias}} = \sum_{j=t+1}^{n} \Big(\mathcal{I} + (\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}(\mathcal{A}_{\mathcal{L}} - \mathcal{A}_{\mathcal{L}}^{n+1-j}) + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top} - (\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j})\Big)\frac{\mathbf{\Phi}_j^{\text{bias}}}{(n-t)^2}$$

Now, we substitute $\mathbf{\Phi}_j^{\text{bias}}$ from equation (B.12):

$$\begin{aligned}
\bar{\mathbf{\Phi}}_{t,n}^{\text{bias}} &= \sum_{j=t+1}^{n} \left(\mathcal{I} + (\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}(\mathcal{A}_{\mathcal{L}} - \mathcal{A}_{\mathcal{L}}^{n+1-j}) + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top} - (\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j})\right)\frac{\mathcal{B}^j\mathbf{\Phi}_0}{(n-t)^2} \\
&= \frac{1}{(n-t)^2}\sum_{j=t+1}^{n}\left(\mathcal{I} + (\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\right)\mathcal{B}^j\mathbf{\Phi}_0 \\
&\quad - \frac{1}{(n-t)^2}\sum_{j=t+1}^{n}\left((\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}}^{n+1-j} + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j}\right)\mathcal{B}^j\mathbf{\Phi}_0 \\
&= \underbrace{\frac{1}{(n-t)^2}\left(\mathcal{I} + (\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\right)(\mathcal{I}-\mathcal{B})^{-1}(\mathcal{B}^{t+1}-\mathcal{B}^{n+1})\mathbf{\Phi}_0}_{\text{Leading order term}} \\
&\quad - \frac{1}{(n-t)^2}\sum_{j=t+1}^{n}\left((\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}}^{n+1-j} + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j}\right)\mathcal{B}^j\mathbf{\Phi}_0. \qquad \text{(B.13)}
\end{aligned}$$

There are two points to note here: (a) The second line consists of terms that constitute the lower-order terms of the bias. We will bound the summation by taking a supremum over $j$. (b) Note that the burn-in phase consisting of $t$ unaveraged iterations allows for a geometric decay of the bias, followed by the tail-averaged phase that allows for a sublinear rate of bias decay. □

### *B.2.3 Covariance of Variance error of the tail-averaged iterate*

*Proof of Lemma 13.* Before obtaining the covariance of the tail-averaged iterate, we note that $\mathbb{E}\left[\boldsymbol{\theta}_j^{\text{variance}}\right] = 0\ \forall\ j$. This can be easily seen since $\boldsymbol{\theta}_0^{\text{variance}} = 0$ and $\mathbb{E}\left[\boldsymbol{\theta}_k^{\text{variance}}\right] = \mathbf{A}\mathbb{E}\left[\boldsymbol{\theta}_{k-1}^{\text{variance}}\right]$ (from equation (B.7)).

Next, in order to obtain the covariance of the variance of the tail-averaged iterate, we first need to obtain $\mathbf{\Phi}_j^{\text{variance}}$, and we will obtain this by unrolling the recursion of equation (B.5):

$$\begin{aligned}
\boldsymbol{\theta}_k^{\text{variance}} &= \widehat{\mathbf{A}}_k\boldsymbol{\theta}_{k-1}^{\text{variance}} + \boldsymbol{\zeta}_k \\
\implies \mathbf{\Phi}_k^{\text{variance}} &= \mathbb{E}\left[\boldsymbol{\theta}_k^{\text{variance}} \otimes \boldsymbol{\theta}_k^{\text{variance}}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\boldsymbol{\theta}_k^{\text{variance}} \otimes \boldsymbol{\theta}_k^{\text{variance}} | \mathcal{F}_{k-1}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\widehat{\mathbf{A}}_k\boldsymbol{\theta}_{k-1}^{\text{variance}} \otimes \boldsymbol{\theta}_{k-1}^{\text{variance}}\widehat{\mathbf{A}}_k^{\top} + \boldsymbol{\zeta}_k \otimes \boldsymbol{\zeta}_k | \mathcal{F}_{k-1}\right]\right]
\end{aligned}$$

$$\begin{aligned}
&= \mathcal{B}\ \mathbb{E}\left[\boldsymbol{\theta}_{k-1}^{\text{variance}} \otimes \boldsymbol{\theta}_{k-1}^{\text{variance}}\right] + \widehat{\boldsymbol{\Sigma}} = \mathcal{B}\ \boldsymbol{\Phi}_{k-1}^{\text{variance}} + \widehat{\boldsymbol{\Sigma}} \\
\implies \boldsymbol{\Phi}_{k}^{\text{variance}} &= \sum_{j=0}^{k-1} \mathcal{B}^j\ \widehat{\boldsymbol{\Sigma}} \\
&= (\mathbf{I} - \mathcal{B})^{-1}(\mathcal{I} - \mathcal{B}^k)\widehat{\boldsymbol{\Sigma}} \qquad \text{(B.14)}
\end{aligned}$$

Note that the cross terms in the outer product computations vanish owing to the fact that $\mathbb{E}\left[\boldsymbol{\theta}_{k-1}^{\text{variance}}\right] = 0\ \forall\ k$. We then recount the expression for the covariance of the variance error from equation (B.11):

$$\bar{\boldsymbol{\Phi}}_{t,n}^{\text{variance}} = \sum_{j=t+1}^{n} \left(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}(\mathcal{A}_{\mathcal{L}} - \mathcal{A}_{\mathcal{L}}^{n+1-j}) + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top} - (\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j})\right)\frac{\boldsymbol{\Phi}_{j}^{\text{variance}}}{(n-t)^2}$$

We will substitute the expression for $\boldsymbol{\Phi}_{j}^{\text{variance}}$ from equation (B.14).

$$\begin{aligned}
\bar{\boldsymbol{\Phi}}_{t,n}^{\text{variance}} = \frac{1}{(n-t)^2}\sum_{j=t+1}^{n} &\left(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}(\mathcal{A}_{\mathcal{L}} - \mathcal{A}_{\mathcal{L}}^{n+1-j}) + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top} - (\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j})\right) \\
&(\mathcal{I} - \mathcal{B})^{-1}(\mathcal{I} - \mathcal{B}^j)\widehat{\boldsymbol{\Sigma}}
\end{aligned}$$

Evaluating this summation, we have:

$$\begin{aligned}
\bar{\boldsymbol{\Phi}}_{t,n}^{\text{variance}} &= \underbrace{\frac{1}{n-t}\big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\big)(\mathcal{I} - \mathcal{B})^{-1}\widehat{\boldsymbol{\Sigma}}}_{\text{Leading order term}} \\
&- \frac{1}{(n-t)^2}\big((\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-2}(\mathcal{A}_{\mathcal{L}} - \mathcal{A}_{\mathcal{L}}^{n+1-t}) + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-2}(\mathcal{A}_{\mathcal{R}}^{\top} - (\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-t})\big)(\mathcal{I} - \mathcal{B})^{-1}\widehat{\boldsymbol{\Sigma}} \\
&- \frac{1}{(n-t)^2}\big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\big)(\mathcal{I} - \mathcal{B})^{-2}(\mathcal{B}^{t+1} - \mathcal{B}^{n+1})\widehat{\boldsymbol{\Sigma}} \\
&+ \frac{1}{(n-t)^2}\sum_{j=t+1}^{n}\big((\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}}^{n+1-j} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j}\big)(\mathcal{I} - \mathcal{B})^{-1}\mathcal{B}^j\widehat{\boldsymbol{\Sigma}}
\end{aligned}$$
(B.15)

□

Equations (B.8), (B.13), (B.15) wrap up the proof of Lemmas 11, 13.

The parameter error of the (tail-)averaged iterate can be obtained using a trace operator $\langle \bullet, \bullet \rangle$ to the tail-averaged iterate's covariance $\bar{\mathbf{\Phi}}_{t,n}$ with the matrix $\begin{bmatrix} \mathbf{I} & 0 \\ 0 & 0 \end{bmatrix}$, i.e.

$$\|\bar{\mathbf{x}}_{t,n} - \mathbf{x}^*\|_2^2 = \langle \begin{bmatrix} \mathbf{I} & 0 \\ 0 & 0 \end{bmatrix}, \bar{\mathbf{\Phi}}_{t,n} \rangle$$

In order to obtain the function error, we note the following taylor expansion of the function $P(\bullet)$ around the minimizer $\mathbf{x}^*$:

$$\begin{aligned} P(\mathbf{x}) &= P(\mathbf{x}^*) + \frac{1}{2} \|\mathbf{x} - \mathbf{x}^*\|^2_{\nabla^2 P(\mathbf{x}^*)} \\ &= P(\mathbf{x}^*) + \frac{1}{2} \|\mathbf{x} - \mathbf{x}^*\|^2_{\mathbf{H}} \end{aligned}$$

This implies the excess risk can be obtained as:

$$\begin{aligned} P(\bar{\mathbf{x}}_{t,n}) - P(\mathbf{x}^*) &= \frac{1}{2} \cdot \langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \bar{\mathbf{\Phi}}_{t,n} \rangle \\ &\leq \langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \bar{\mathbf{\Phi}}^{\text{bias}}_{t,n} \rangle + \langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \bar{\mathbf{\Phi}}^{\text{variance}}_{t,n} \rangle \end{aligned}$$

## *B.3 Useful lemmas*

In this section, we will state and prove some useful lemmas that will be helpful in the later sections.

**Lemma 31.**

$$\left(\mathbf{I} - \mathbf{A}^\top\right)^{-1} \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix} = \frac{1}{q - c\delta} \begin{bmatrix} -(c\mathbf{I} - q\mathbf{H}) & 0 \\ (\mathbf{I} - \delta\mathbf{H}) & 0 \end{bmatrix}$$

*Proof.* Since we assumed that $\mathbf{H}$ is a diagonal matrix (with out loss of generality), we note that $\mathbf{A}$ is a block diagonal matrix after a rearrangement of the co-ordinates (via an eigenvalue decomposition).

In particular, by considering the $j^{\text{th}}$ block (denoted by $\mathbf{A}_j$ corresponding to the $j^{\text{th}}$ eigenvalue $\lambda_j$ of $\mathbf{H}$), we have:

$$\mathbf{I} - \mathbf{A}_j^\top = \begin{bmatrix} 1 & c \\ -(1-\delta\lambda_j) & -(c - q\lambda_j) \end{bmatrix}$$

Implying that the determinant $\left|\mathbf{I} - \mathbf{A}_j^\top\right| = (q - c\delta)\lambda_j$, using which:

$$(\mathbf{I} - \mathbf{A}_j^\top)^{-1} = \frac{1}{(q - c\delta)\lambda_j} \begin{bmatrix} -(c - q\lambda_j) & -c \\ 1 - \delta\lambda_j & 1 \end{bmatrix} \tag{B.16}$$

Thus,

$$(\mathbf{I} - \mathbf{A}_j^\top)^{-1} \begin{bmatrix} \lambda_j & 0 \\ 0 & 0 \end{bmatrix} = \frac{1}{q - c\delta} \begin{bmatrix} -(c - q\lambda_j) & 0 \\ (1 - \delta\lambda_j) & 0 \end{bmatrix}$$

Accumulating the results of each of the blocks and by rearranging the co-ordinates, the result follows. □

**Lemma 32.**

$$\left(\mathbf{I} - \mathbf{A}^\top\right)^{-1} \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix} (\mathbf{I} - \mathbf{A})^{-1} = \frac{1}{(q - c\delta)^2} \left( \otimes_2 \begin{bmatrix} -(c\mathbf{I} - q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I} - \delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix} \right)$$

*Proof.* In a similar manner as in Lemma 31, we decompose the computation into each of the eigen-directions and subsequently re-arrange the results. In particular, we note:

$$(\mathbf{I} - \mathbf{A}_j)^{-1} = \frac{1}{(q - c\delta)\lambda_j} \begin{bmatrix} -(c - q\lambda_j) & (1 - \delta\lambda_j) \\ -c & 1 \end{bmatrix}$$

Multiplying the above with the result of Lemma 31, we have:

$$(\mathbf{I} - \mathbf{A}_j^\top)^{-1} \begin{bmatrix} \lambda_j & 0 \\ 0 & 0 \end{bmatrix} (\mathbf{I} - \mathbf{A}_j)^{-1} = \frac{1}{(q - c\delta)^2} \left( \otimes_2 \begin{bmatrix} -(c - q\lambda_j)\lambda_j^{-1/2} \\ (1 - \delta\lambda_j)\lambda_j^{-1/2} \end{bmatrix} \right)$$

From which the statement of the lemma follows through a simple re-arrangement.

□

**Lemma 33.**

$$\left(\mathbf{I}-\mathbf{A}^\top\right)^{-2}\mathbf{A}^\top\begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix}=\frac{1}{(q-c\delta)^2}\begin{bmatrix}\mathbf{H}^{-1}(-c(1-c)\mathbf{I}-cq\mathbf{H})(\mathbf{I}-\delta\mathbf{H}) & 0\\ \mathbf{H}^{-1}((1-c)\mathbf{I}-c\delta\mathbf{H})(\mathbf{I}-\delta\mathbf{H}) & 0\end{bmatrix}$$

*Proof.* In a similar argument as in previous two lemmas, we analyze the expression in each eigendirection of $\mathbf{H}$ through a rearrangement of the co-ordinates. Utilizing the expression of $\mathbf{I}-\mathbf{A}_j^\top$ from equation (B.16), we get:

$$(\mathbf{I}-\mathbf{A}_j^\top)^{-1}\mathbf{A}_j^\top\begin{bmatrix}\lambda_j & 0\\ 0 & 0\end{bmatrix}=\frac{1}{(q-c\delta)}\begin{bmatrix}-c(1-\delta\lambda_j) & 0\\ (1-\delta\lambda_j) & 0\end{bmatrix} \tag{B.17}$$

thus implying:

$$(\mathbf{I}-\mathbf{A}_j^\top)^{-2}\mathbf{A}_j^\top\begin{bmatrix}\lambda_j & 0\\ 0 & 0\end{bmatrix}=\frac{(1-\delta\lambda_j)}{(q-c\delta)^2\lambda_j}\begin{bmatrix}-c(1-c)-cq\lambda_j & 0\\ (1-c)-c\delta\lambda_j & 0\end{bmatrix}$$

Rearranging the co-ordinates, the statement of the lemma follows. □

**Lemma 34.** *The matrix* $\mathbf{A}$ *satisfies the following properties:*

1. *Eigenvalues $q$ of $\mathbf{A}$ satisfy $|q|\le\sqrt{\alpha}$, and*

2. $\left\|\mathbf{A}^k\right\|_2\le 3\sqrt{2}\cdot k\cdot\alpha^{\frac{k-1}{2}}\ \forall\, k\ge 1.$

*Proof.* Since the matrix is block-diagonal with $2\times 2$ blocks, after a rearranging the coordinates, we will restrict ourselves to bounding the eigenvalues and eigenvectors of each of these $2\times 2$ blocks. Combining the results for different blocks then proves the lemma. Recall that $\mathbf{A}_j=\begin{bmatrix}0 & 1-\delta\lambda_j\\ -c & 1+c-q\lambda_j\end{bmatrix}$.

**Part I**: Let us first prove the statement about the eigenvalues of $\mathbf{A}$. There are two scenarios here:

1. *Complex eigenvalues*: In this case, both eigenvalues of $\mathbf{A}_j$ have the same magnitude which is given by $\sqrt{\det(\mathbf{A}_j)}=\sqrt{c(1-\delta\lambda_j)}\le\sqrt{c}\le\sqrt{\alpha}$.

2. *Real eigenvalues*: Let $q_1$ and $q_2$ be the two real eigenvalues of $\mathbf{A}_j$. We know that $q_1 + q_2 = \mathrm{Tr}(\mathbf{A}_j) = 1 + c - q\lambda_j > 0$ and $q_1 \cdot q_2 = \det(\mathbf{A}_j) > 0$. This means that $q_1 > 0$ and $q_2 > 0$. Now, consider the matrix $\mathbf{G}_j \overset{\text{def}}{=} (1-\beta)\mathbf{I} - \mathbf{A}_j = \begin{bmatrix} (1-\beta) & -1+\delta\lambda_j \\ c & -1+(1-\beta)(1-\alpha)+q\lambda_j \end{bmatrix}$. We see that $((1-\beta)-q_1)((1-\beta)-q_2) = \det(\mathbf{G}_j) = (1-\beta)(1-\alpha)\left((1-\beta)-1\right)+(1-\beta)\left(q-\alpha\delta\right)\lambda_j = (1-\beta)\left(1-\alpha\right)\left(\gamma\lambda_j - \beta\right) \geq 0$. This means that there are two possibilities: either $q_1, q_2 \geq (1-\beta)$ or $q_1, q_2 \leq (1-\beta)$. If the second condition is true, then we are done. If not, if $q_1, q_2 \geq (1-\beta)$, then $\max_i q_i = \frac{\det(\mathbf{A}_j)}{\min_i q_i} \leq \frac{c(1-\delta\lambda_j)}{(1-\beta)} \leq \alpha(1-\delta\lambda_j)$. Since $\sqrt{\alpha} \geq \alpha \geq 1-\beta$, this proves the first part of the lemma.

**Part II**: Let $\mathbf{A}_j = \mathbf{V}\mathbf{Q}\mathbf{V}^\top$ be the Schur decomposition of $\mathbf{A}_j$ where $\mathbf{Q} = \begin{bmatrix} q_1 & q \\ 0 & q_2 \end{bmatrix}$ is an upper triangular matrix with eigenvalues $q_1$ and $q_2$ of $\mathbf{A}_j$ on the diagonal and $\mathbf{V}$ is a unitary matrix i.e., $\mathbf{V}\mathbf{V}^\top = \mathbf{V}^\top\mathbf{V} = \mathbf{I}$. We first observe that $|q| \leq \|\mathbf{Q}\|_2 \overset{(\zeta_1)}{=} \|\mathbf{A}_j\|_2 \leq \|\mathbf{A}_j\|_F \leq \sqrt{6}$, where $(\zeta_1)$ follows from the fact that $\mathbf{V}$ is a unitary matrix. $\mathbf{V}$ being unitary also implies that $\mathbf{A}_j^k = \mathbf{V}\mathbf{Q}^k\mathbf{V}^\top$. On the other hand, a simple proof via induction tells us that

$$\mathbf{Q}^k = \begin{bmatrix} q_1^k & q\left(\sum_{\ell=1}^{k-1} q_1^\ell q_2^{k-\ell}\right) \\ 0 & q_2^k \end{bmatrix}.$$

So, we have $\left\|\mathbf{A}_j^k\right\|_2 = \left\|\mathbf{Q}^k\right\|_2 \leq \left\|\mathbf{Q}^k\right\|_F \leq \sqrt{3}k\,|q| \max\left(|q_1|^{k-1}, |q_2|^{k-1}\right) \leq 3\sqrt{2}\cdot k \cdot \alpha^{\frac{k-1}{2}}$, where we used $|q| \leq \sqrt{6}$ and $\max\left(|q_1|, |q_2|\right) \leq \sqrt{\alpha}$. □

Finally, we state and prove the following lemma which is a relation between left and right multiplication operators.

**Lemma 35.** *Let* $\mathbf{A}$ *be any matrix with* $\mathcal{A}_\mathcal{L} = \mathbf{A} \otimes \mathbf{I}$ *and* $\mathcal{A}_\mathcal{R} = \mathbf{I} \otimes \mathbf{A}$ *representing its left and right multiplication operators. Then, the following expression holds:*

$$\left(\mathcal{I} + (\mathcal{I} - \mathcal{A}_\mathcal{L})^{-1}\mathcal{A}_\mathcal{L} + (\mathcal{I} - \mathcal{A}_\mathcal{R}^\top)^{-1}\mathcal{A}_\mathcal{R}^\top\right)(\mathcal{I} - \mathcal{A}_\mathcal{L}\mathcal{A}_\mathcal{R}^\top)^{-1} = (\mathcal{I} - \mathcal{A}_\mathcal{L})^{-1}(\mathcal{I} - \mathcal{A}_\mathcal{R}^\top)^{-1}$$

*Proof.* Let us assume that $\mathbf{A}$ can be written in terms of its eigen decomposition as $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^{-1}$. Then the first claim is that $\mathcal{I}, \mathcal{A}_\mathcal{L}, \mathcal{A}_\mathcal{R}$ are diagonalized by the same basis consisting of the eigenvectors of $\mathbf{A}$, i.e. in particular, the matrix of eigenvectors of $\mathcal{I}, \mathcal{A}_\mathcal{L}, \mathcal{A}_\mathcal{R}$ can be written as $\mathbf{V} \otimes \mathbf{V}$. In particular, this implies, $\forall\ i, j \in \{1, 2, ..., d\} \times \{1, 2, ..., d\}$, we have, applying $\mathbf{v}_i \otimes \mathbf{v}_j$ to the LHS, we have:

$$\begin{aligned}
&\left(\mathcal{I} + (\mathcal{I} - \mathcal{A}_\mathcal{L})^{-1}\mathcal{A}_\mathcal{L} + (\mathcal{I} - \mathcal{A}_\mathcal{R}^\top)^{-1}\mathcal{A}_\mathcal{R}^\top\right)(\mathcal{I} - \mathcal{A}_\mathcal{L}\mathcal{A}_\mathcal{R}^\top)^{-1}\mathbf{v}_i \otimes \mathbf{v}_j \\
&= (1 - \lambda_i\lambda_j)^{-1}\left(\mathcal{I} + (\mathcal{I} - \mathcal{A}_\mathcal{L})^{-1}\mathcal{A}_\mathcal{L} + (\mathcal{I} - \mathcal{A}_\mathcal{R}^\top)^{-1}\mathcal{A}_\mathcal{R}^\top\right)\mathbf{v}_i \otimes \mathbf{v}_j \\
&= (1 + \lambda_i(1 - \lambda_i)^{-1} + \lambda_j(1 - \lambda_j)^{-1}) \cdot (1 - \lambda_i\lambda_j)^{-1}\mathbf{v}_i \otimes \mathbf{v}_j
\end{aligned}$$

Applying $\mathbf{v}_i \otimes \mathbf{v}_j$ to the RHS, we have:

$$\begin{aligned}
&(\mathcal{I} - \mathcal{A}_\mathcal{L})^{-1}(\mathcal{I} - \mathcal{A}_\mathcal{R}^\top)^{-1}\mathbf{v}_i \otimes \mathbf{v}_j \\
&= (1 - \lambda_i)^{-1}(1 - \lambda_j)^{-1}\mathbf{v}_i \otimes \mathbf{v}_j
\end{aligned}$$

The next claim is that for any scalars (real/complex) $x, y \neq 1$, the following statement holds implying the statement of the lemma:

$$(1 + (1 - x)^{-1}x + (1 - y)^{-1}y) \cdot (1 - xy)^{-1} = (1 - x)^{-1}(1 - y)^{-1}$$

□

**Lemma 36.** *Recall the matrix* $\mathbf{G}$ *defined as* $\mathbf{G} \overset{def}{=} \begin{bmatrix} \mathbf{I} & \frac{-\alpha}{1-\alpha}\mathbf{I} \\ 0 & \frac{1}{1-\alpha}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mu\mathbf{H}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & 0 \\ \frac{-\alpha}{1-\alpha}\mathbf{I} & \frac{1}{1-\alpha}\mathbf{I} \end{bmatrix}$. *The condition number of* $\mathbf{G}$, $\kappa(\mathbf{G})$ *satisfies* $\kappa(\mathbf{G}) \leq \frac{4\kappa}{\sqrt{1-\alpha^2}}$.

*Proof.* Since the above matrix is block-diagonal after a rearrangement of coordinates, it suffices to compute the smallest and largest singular values of each block. Let $\lambda_i$ be the $i^{\text{th}}$ eigenvalue of $\mathbf{H}$. Let $\mathbf{C} \overset{\text{def}}{=} \begin{bmatrix} 1 & 0 \\ \frac{-\alpha}{1-\alpha} & \frac{1}{1-\alpha} \end{bmatrix}$ and consider the matrix $\mathbf{G}_i \overset{\text{def}}{=} \mathbf{C} \begin{bmatrix} 1 & 0 \\ 0 & \frac{\mu}{\lambda_i} \end{bmatrix} \mathbf{C}^\top$. The largest eigenvalue of $\mathbf{G}_i$ is at most $\sigma_{\max}(\mathbf{C})^2$, while the smallest eigenvalue, $\sigma_{\min}(\mathbf{G}_i)$ is at least $\frac{\mu}{\lambda_i} \cdot \sigma_{\min}(\mathbf{C})^2$. We obtain the following bounds on $\sigma_{\min}(\mathbf{C})$ and $\sigma_{\max}(\mathbf{C})$.

$$\sigma_{\max}(\mathbf{C}) \leq \|\mathbf{C}\|_F \leq \frac{2}{\sqrt{1-\alpha^2}} \quad (\because\ \alpha \leq 1)$$

$$\sigma_{\min}(\mathbf{C}) \geq \frac{\sqrt{\det(\mathbf{C}\mathbf{C}^\top)}}{\|\mathbf{C}\|_F} \geq \frac{1}{2},$$
$$\left(\because \det\left(\mathbf{C}\mathbf{C}^\top\right) = \sigma_{\max}(\mathbf{C})^2 \sigma_{\min}(\mathbf{C})^2\right)$$

where we used the computation that $\det\left(\mathbf{C}\mathbf{C}^\top\right) = \frac{1}{1-\alpha}$. This means that $\sigma_{\min}(\mathbf{G}_i) \geq \frac{\mu}{2\lambda_i}$ and $\sigma_{\max}(\mathbf{G}_i) \leq \frac{2}{\sqrt{1-\alpha^2}}$. Combining all the blocks, we see that the condition number of $\mathbf{G}$ is at most $\frac{4\kappa}{\sqrt{1-\alpha^2}}$, proving the lemma. □

## B.4 Lemmas and proofs for bias contraction

*Proof of Lemma 12.* Let $\mathbf{v} \overset{\text{def}}{=} \frac{1}{1-\alpha}(\mathbf{y} - \alpha\mathbf{x})$ and consider the following update rules corresponding to the noiseless versions of the updates in Algorithm 3:

$$\begin{aligned}
\mathbf{x}^+ &= \mathbf{y} - \delta\widehat{\mathbf{H}}(\mathbf{y} - \mathbf{x}^*)\\
\mathbf{z} &= \beta\mathbf{y} + (1-\beta)\mathbf{v}\\
\mathbf{v}^+ &= \mathbf{z} - \gamma\widehat{\mathbf{H}}(\mathbf{y} - \mathbf{x}^*)\\
\mathbf{y}^+ &= \alpha\mathbf{x}^+ + (1-\alpha)\mathbf{v}^+,
\end{aligned}$$

where $\widehat{\mathbf{H}} \overset{\text{def}}{=} \mathbf{a}\mathbf{a}^\top$ where $\mathbf{a}$ is sampled from the marginal on $(\mathbf{a}, b) \sim \mathcal{D}$. We first note that

$$\begin{aligned}
\mathbb{E}\left[\otimes_2 \begin{bmatrix}\mathbf{x}^+ - \mathbf{x}^*\\ \mathbf{y}^+ - \mathbf{x}^*\end{bmatrix}\right] &= \mathbb{E}\left[\widehat{\mathbf{A}}\left(\otimes_2 \begin{bmatrix}\mathbf{x} - \mathbf{x}^*\\ \mathbf{y} - \mathbf{x}^*\end{bmatrix}\right)\widehat{\mathbf{A}}^\top\right]\\
&= \mathcal{B}\left(\otimes_2 \begin{bmatrix}\mathbf{x} - \mathbf{x}^*\\ \mathbf{y} - \mathbf{x}^*\end{bmatrix}\right)
\end{aligned}$$

Letting $\widetilde{\mathbf{G}} \overset{\text{def}}{=} \begin{bmatrix}\mathbf{I} & 0\\ \frac{-\alpha}{1-\alpha}\mathbf{I} & \frac{1}{1-\alpha}\mathbf{I}\end{bmatrix}$, we can verify that $\begin{bmatrix}\mathbf{x} - \mathbf{x}^*\\ \mathbf{v} - \mathbf{x}^*\end{bmatrix} = \widetilde{\mathbf{G}}\begin{bmatrix}\mathbf{x} - \mathbf{x}^*\\ \mathbf{y} - \mathbf{x}^*\end{bmatrix}$, similarly $\begin{bmatrix}\mathbf{x}^+ - \mathbf{x}^*\\ \mathbf{v}^+ - \mathbf{x}^*\end{bmatrix} = \widetilde{\mathbf{G}}\begin{bmatrix}\mathbf{x}^+ - \mathbf{x}^*\\ \mathbf{y}^+ - \mathbf{x}^*\end{bmatrix}$. Recall that $\mathbf{G} \overset{\text{def}}{=} \widetilde{\mathbf{G}}^\top \begin{bmatrix}\mathbf{I} & 0\\ 0 & \mu\mathbf{H}^{-1}\end{bmatrix}\widetilde{\mathbf{G}}$. With this notation in place, we prove the statement below, and substitute the values of $c_1, c_2, c_3$ to obtain the statement of the lemma:

$$\langle \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mu \cdot \mathbf{H}^{-1} \end{bmatrix}, \otimes_2 \Big( \begin{bmatrix} \mathbf{x}^+ - \mathbf{x}^* \\ \mathbf{v}^+ - \mathbf{x}^* \end{bmatrix} \Big) \rangle \leq \left( 1 - c_3 \frac{c_2 \sqrt{2c_1 - c_1^2}}{\sqrt{\kappa \widetilde{\kappa}}} \right) \cdot \langle \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mu \cdot \mathbf{H}^{-1} \end{bmatrix}, \otimes_2 \Big( \begin{bmatrix} \mathbf{x} - \mathbf{x}^* \\ \mathbf{v} - \mathbf{x}^* \end{bmatrix} \Big) \rangle \tag{B.18}$$

To establish this result, let us define two quantities: $e \overset{\text{def}}{=} \|\mathbf{x} - \mathbf{x}^*\|_2^2$, $f \overset{\text{def}}{=} \|\mathbf{v} - \mathbf{x}^*\|_{\mathbf{H}^{-1}}^2$ and similarly, $e^+ \overset{\text{def}}{=} \|\mathbf{x}^+ - \mathbf{x}^*\|_2^2$ and $f^+ \overset{\text{def}}{=} \|\mathbf{v}^+ - \mathbf{x}^*\|_{\mathbf{H}^{-1}}^2$. The potential function we consider is $e + \mu \cdot f$. Recall that the parameters are chosen as:

$$\alpha = \frac{\sqrt{\kappa \widetilde{\kappa}}}{c_2 \sqrt{2c_1 - c_1^2} + \sqrt{\kappa \widetilde{\kappa}}}, \quad \beta = c_3 \frac{c_2 \sqrt{2c_1 - c_1^2}}{\sqrt{\kappa \widetilde{\kappa}}}, \quad \gamma = c_2 \frac{\sqrt{2c_1 - c_1^2}}{\mu \sqrt{\kappa \widetilde{\kappa}}}, \quad \delta = \frac{c_1}{R^2}$$

with $c_1 < 1/2$, $c_3 = \frac{c_2 \sqrt{2c_1 - c_1^2}}{c_1}$, $c_2^2 = \frac{c_4}{2 - c_1}$. Consider $e^+$ and employ the simple gradient descent bound:

$$\begin{aligned} e^+ = \mathbb{E}\left[ \left\| \mathbf{x}^+ - \mathbf{x}^* \right\|_2^2 \right] &= \mathbb{E}\left[ \left\| \mathbf{y} - \delta \cdot \widehat{\mathbf{H}}(\mathbf{y} - \mathbf{x}^*) - \mathbf{x}^* \right\|_2^2 \right] \\ &= \mathbb{E}\left[ \|\mathbf{y} - \mathbf{x}^*\|_2^2 \right] - 2\delta \cdot \mathbb{E}\left[ \|\mathbf{y} - \mathbf{x}^*\|_{\mathbf{H}}^2 \right] + \delta^2 \mathbb{E}\left[ \|\mathbf{y} - \mathbf{x}^*\|_{\mathcal{M}\mathbf{I}}^2 \right] \\ &\leq \mathbb{E}\left[ \|\mathbf{y} - \mathbf{x}^*\|_2^2 \right] - 2\delta \cdot \mathbb{E}\left[ \|\mathbf{y} - \mathbf{x}^*\|_{\mathbf{H}}^2 \right] + R^2 \delta^2 \mathbb{E}\left[ \|\mathbf{y} - \mathbf{x}^*\|_{\mathbf{H}}^2 \right] \\ &= \mathbb{E}\left[ \|\mathbf{y} - \mathbf{x}^*\|_2^2 \right] - \frac{2c_1 - c_1^2}{R^2} \mathbb{E}\left[ \|\mathbf{y} - \mathbf{x}^*\|_{\mathbf{H}}^2 \right] \end{aligned} \tag{B.19}$$

Next, consider $f^+$:

$$\begin{aligned} f^+ = \mathbb{E}\left[ \left\| \mathbf{v}^+ - \mathbf{x}^* \right\|_{\mathbf{H}^{-1}}^2 \right] &= \mathbb{E}\left[ \left\| \mathbf{z} - \gamma \widehat{\mathbf{H}}(\mathbf{y} - \mathbf{x}^*) - \mathbf{x}^* \right\|_{\mathbf{H}^{-1}}^2 \right] \\ &= \mathbb{E}\left[ \|\mathbf{z} - \mathbf{x}^*\|_{\mathbf{H}^{-1}}^2 \right] + \gamma^2 \mathbb{E}\left[ \|\mathbf{y} - \mathbf{x}^*\|_{\mathcal{M}\mathbf{H}^{-1}}^2 \right] - 2\gamma \mathbb{E}\left[ \langle \mathbf{z} - \mathbf{x}^*, \mathbf{y} - \mathbf{x}^* \rangle \right] \\ &\leq \mathbb{E}\left[ \|\mathbf{z} - \mathbf{x}^*\|_{\mathbf{H}^{-1}}^2 \right] + \gamma^2 \widetilde{\kappa} \cdot \mathbb{E}\left[ \|\mathbf{y} - \mathbf{x}^*\|_{\mathbf{H}}^2 \right] - 2\gamma \cdot \mathbb{E}\left[ \langle \mathbf{z} - \mathbf{x}^*, \mathbf{y} - \mathbf{x}^* \rangle \right] \end{aligned} \tag{B.20}$$

Where, we use the fact that $\mathcal{M}\mathbf{H}^{-1} \preceq \widetilde{\kappa}\mathbf{H}$, where $\widetilde{\kappa}$ is the *statistical* condition number. Consider $\mathbb{E}\left[ \|\mathbf{z} - \mathbf{x}^*\|_{\mathbf{H}^{-1}}^2 \right]$ and use convexity of the weighted 2−norm to get:

$$\mathbb{E}\left[ \|\mathbf{z} - \mathbf{x}^*\|_{\mathbf{H}^{-1}}^2 \right] \leq \beta \mathbb{E}\left[ \|\mathbf{y} - \mathbf{x}^*\|_{\mathbf{H}^{-1}}^2 \right] + (1 - \beta) \mathbb{E}\left[ \|\mathbf{v} - \mathbf{x}^*\|_{\mathbf{H}^{-1}}^2 \right]$$

$$\leq \frac{\beta}{\mu}\mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_2^2\right] + (1-\beta)\cdot f \tag{B.21}$$

Next, consider $\mathbb{E}\left[\langle \mathbf{z}-\mathbf{x}^*, \mathbf{y}-\mathbf{x}^*\rangle\right]$, and first write $\mathbf{z}$ in terms of $\mathbf{x}$ and $\mathbf{y}$. This can be seen as two steps:

- $\mathbf{v} = \frac{1}{1-\alpha}\cdot\mathbf{y} - \frac{\alpha}{1-\alpha}\cdot\mathbf{x}$
- $\mathbf{z} = \beta\mathbf{y} + (1-\beta)\mathbf{v} = \mathbf{y} + (1-\beta)(\mathbf{v}-\mathbf{y})$. Then substituting $\mathbf{v}$ in terms of $\mathbf{x}$ and $\mathbf{y}$ as in the equation above, we get: $\mathbf{z} = \mathbf{y} + \left(\frac{\alpha\cdot(1-\beta)}{1-\alpha}\right)(\mathbf{y}-\mathbf{x})$

Then, $\mathbb{E}\left[\langle \mathbf{z}-\mathbf{x}^*, \mathbf{y}-\mathbf{x}^*\rangle\right]$ can be written as:

$$\mathbb{E}\left[\langle \mathbf{z}-\mathbf{x}^*, \mathbf{y}-\mathbf{x}^*\rangle\right] = \mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_2^2\right] + \left(\frac{\alpha(1-\beta)}{1-\alpha}\right)\mathbb{E}\left[\langle \mathbf{y}-\mathbf{x}, \mathbf{y}-\mathbf{x}^*\rangle\right] \tag{B.22}$$

Then, we note:

$$\begin{aligned}
\mathbb{E}\left[\langle \mathbf{y}-\mathbf{x}, \mathbf{y}-\mathbf{x}^*\rangle\right] &= \mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_2^2\right] - \mathbb{E}\left[\langle \mathbf{x}-\mathbf{x}^*, \mathbf{y}-\mathbf{x}^*\rangle\right] \\
&\geq \mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_2^2\right] - \frac{1}{2}\cdot\left(\mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_2^2\right] + \mathbb{E}\left[\|\mathbf{x}-\mathbf{x}^*\|_2^2\right]\right) \\
&= \frac{1}{2}\cdot\left(\mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_2^2\right] - \mathbb{E}\left[\|\mathbf{x}-\mathbf{x}^*\|_2^2\right]\right)
\end{aligned}$$

Re-substituting in equation (B.22):

$$\begin{aligned}
\mathbb{E}\left[\langle \mathbf{z}-\mathbf{x}^*, \mathbf{y}-\mathbf{x}^*\rangle\right] &\geq \left(1 + \frac{1}{2}\cdot\frac{\alpha(1-\beta)}{1-\alpha}\right)\mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_2^2\right] - \frac{1}{2}\cdot\frac{\alpha(1-\beta)}{1-\alpha}\mathbb{E}\left[\|\mathbf{x}-\mathbf{x}^*\|_2^2\right] \\
&= \left(1 + \frac{1}{2}\cdot\frac{\alpha(1-\beta)}{1-\alpha}\right)\mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_2^2\right] - \frac{1}{2}\cdot\frac{\alpha(1-\beta)}{1-\alpha}\cdot e \qquad \text{(B.23)}
\end{aligned}$$

Substituting equations (B.21), (B.23) into equation (B.20), we get:

$$\begin{aligned}
\mu\cdot f^+ \leq &\left(\beta - 2\gamma\mu - \frac{\gamma\mu\alpha(1-\beta)}{1-\alpha}\right)\mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_2^2\right] + \mu(1-\beta)\cdot f \\
&+ \frac{\gamma\mu\alpha(1-\beta)}{1-\alpha}\cdot e + \mu\gamma^2\widetilde{\kappa}\cdot\mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_{\mathbf{H}}^2\right]
\end{aligned}$$

Rewriting the guarantee on $e^+$ as in equation (B.19):

$$e^+ \leq \mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_2^2\right] - \frac{2c_1 - c_1^2}{R^2}\cdot\mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_{\mathbf{H}}^2\right]$$

By considering $e^+ + \mu\cdot f^+$, we see the following:

- The coefficient of $\mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_{\mathbf{H}}^2\right] \leq 0$ by setting $\gamma = c_2 \frac{\sqrt{2c_1 - c_1^2}}{\mu\sqrt{\kappa\widetilde{\kappa}}}$, where, $0 < c_2 \leq 1$, $\kappa = \frac{R^2}{\mu}$.

- Set $\frac{\gamma\mu\alpha}{1-\alpha} = 1$ implying $\alpha = \frac{1}{1+\gamma\mu} = \frac{\sqrt{\kappa\widetilde{\kappa}}}{c_2\sqrt{2c_1-c_1^2}+\sqrt{\kappa\widetilde{\kappa}}}$

With these in place, we have the final result:

$$e^+ + \mu \cdot f^+ \leq (2\beta - 2\gamma\mu)\mathbb{E}\left[\|\mathbf{y}-\mathbf{x}^*\|_2^2\right] + (1-\beta)\cdot(e + \mu\cdot f)$$

In particular, setting $\beta = c_3\gamma\mu = c_3\frac{c_2\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}$, we have a per-step contraction of $1-\beta$ which is precisely $1 - c_3\frac{c_2\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}$, from which the claimed result naturally follows by substituting the values of $c_1, c_2, c_3$. □

**Lemma 37.** *For any psd matrix* $\mathbf{Q} \succeq 0$*, we have:*

$$\left\|\mathcal{B}^k\mathbf{Q}\right\|_2 \leq \frac{4\kappa}{\sqrt{1-\alpha^2}}\left(1 - \left(\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right)\right)^k \|\mathbf{Q}\|_2 .$$

*Proof.* From Lemma 12, we conclude that $\langle\mathbf{G}, \mathcal{B}^k\mathbf{Q}\rangle \leq \left(1 - \left(\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right)\right)^k \langle\mathbf{G},\mathbf{Q}\rangle$. This implies that $\left\|\mathcal{B}^k\mathbf{Q}\right\|_2 \leq \left(1 - \left(\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right)\right)^k \|\mathbf{Q}\|_2\,\kappa(\mathbf{G})$. Plugging the bound on $\kappa(\mathbf{G})$ from Lemma 36 proves the lemma. □

**Lemma 38.** *We have:*

$$\begin{aligned}&(\mathbf{I}-\mathcal{D})\,(\mathbf{I}-\mathcal{B})^{-1}\mathcal{B}^{t+1}\left(\mathbf{I}-\mathcal{B}^{n-t}\right)\boldsymbol{\theta}_0\boldsymbol{\theta}_0^\top \\ &\preceq \frac{4\kappa}{\sqrt{1-\alpha^2}}\exp\left(-tc_2c_3\sqrt{2c_1-c_1^2}/\sqrt{\kappa\widetilde{\kappa}}\right)\|\boldsymbol{\theta}_0\|^2\left(\mathbf{I} + \frac{\sqrt{\kappa\widetilde{\kappa}}}{c_2c_3\sqrt{2c_1-c_1^2}}(R^2/\sigma^2)\widehat{\boldsymbol{\Sigma}}\right).\end{aligned}$$

*Proof.* The proof follows from Lemma 12. Since $\mathcal{B} = \mathcal{D} + \mathcal{R}$, we have $(\mathcal{I}-\mathcal{D})\,(\mathcal{I}-\mathcal{B})^{-1} = \mathcal{I} + \mathcal{R}(\mathcal{I}-\mathcal{B})^{-1}$. Since $\mathcal{R}, \mathcal{B}$ and $(\mathcal{I}-\mathcal{B})^{-1}$ are all PSD operators, we have

$$\begin{aligned}&(\mathcal{I}-\mathcal{D})\,(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{t+1}\left(\mathcal{I}-\mathcal{B}^{n-t}\right)\boldsymbol{\theta}_0\boldsymbol{\theta}_0^\top \\ &= \left(\mathcal{I} + \mathcal{R}(\mathcal{I}-\mathcal{B})^{-1}\right)\mathcal{B}^{t+1}\left(\mathcal{I}-\mathcal{B}^{n-t}\right)\boldsymbol{\theta}_0\boldsymbol{\theta}_0^\top\end{aligned}$$

$$\preceq \underbrace{\mathcal{B}^{t+1}\boldsymbol{\theta}_0\boldsymbol{\theta}_0^\top}_{\mathbf{S}_1 \overset{\text{def}}{=}} + \underbrace{\mathcal{R}(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{t+1}\boldsymbol{\theta}_0\boldsymbol{\theta}_0^\top}_{\mathbf{S}_2 \overset{\text{def}}{=}}.$$

Applying Lemma 37 with $\mathbf{Q} = \boldsymbol{\theta}_0\boldsymbol{\theta}_0^\top$ tells us that $\mathbf{S}_1 \preceq \frac{4\kappa}{\sqrt{1-\alpha^2}} \exp\left(-tc_2c_3\sqrt{2c_1-c_1^2}/\sqrt{\kappa\widetilde{\kappa}}\right)\|\boldsymbol{\theta}_0\|_2^2\,\mathbf{I}$. For $\mathbf{S}_2$, we have

$$\begin{aligned}\langle \mathbf{G}, (\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{t+1}\boldsymbol{\theta}_0\boldsymbol{\theta}_0^\top\rangle &= \langle \mathbf{G}, \sum_{j=t+1}^{\infty}\mathcal{B}^j\boldsymbol{\theta}_0\boldsymbol{\theta}_0^\top\rangle \\ &\le \sum_{j=t+1}^{\infty}\left(1-\left(\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right)\right)^j \langle \mathbf{G}, \boldsymbol{\theta}_0\boldsymbol{\theta}_0^\top\rangle \\ &\le \frac{\sqrt{\kappa\widetilde{\kappa}}}{c_2c_3\sqrt{2c_1-c_1^2}}\exp\left(-tc_2c_3\sqrt{2c_1-c_1^2}/\sqrt{4\kappa\widetilde{\kappa}}\right)\langle \mathbf{G}, \boldsymbol{\theta}_0\boldsymbol{\theta}_0^\top\rangle.\end{aligned}$$

This implies

$$(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{t+1}\boldsymbol{\theta}_0\boldsymbol{\theta}_0^\top \preceq \kappa(\mathbf{G})(\sqrt{\kappa\widetilde{\kappa}}/(c_2c_3\sqrt{2c_1-c_1^2}))\exp\left(-tc_2c_3\sqrt{2c_1-c_1^2}/\sqrt{4\kappa\widetilde{\kappa}}\right)\|\boldsymbol{\theta}_0\|^2\,\mathbf{I},$$

which tells us that

$$\mathbf{S}_2 \preceq \kappa(\mathbf{G})(\sqrt{\kappa\widetilde{\kappa}}/(c_2c_3\sqrt{2c_1-c_1^2}))\exp\left(-tc_2c_3\sqrt{2c_1-c_1^2}/\sqrt{4\kappa\widetilde{\kappa}}\right)\|\boldsymbol{\theta}_0\|^2(R^2/\sigma^2)\widehat{\boldsymbol{\Sigma}}$$

Combining the bounds on $\mathbf{S}_1$ and $\mathbf{S}_2$, we obtain

$$\begin{aligned}&(\mathcal{I}-\mathcal{D})(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{t+1}\left(\mathcal{I}-\mathcal{B}^{n-t}\right)\boldsymbol{\theta}_0\boldsymbol{\theta}_0^\top \\ &\preceq \kappa(\mathbf{G})\exp\left(-tc_2c_3\sqrt{2c_1-c_1^2}/\sqrt{4\kappa\widetilde{\kappa}}\right)\|\boldsymbol{\theta}_0\|^2\left(\mathbf{I}+\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_2c_3\sqrt{2c_1-c_1^2}}(R^2/\sigma^2)\widehat{\boldsymbol{\Sigma}}\right).\end{aligned}$$

Plugging the bound for $\kappa(\mathbf{G})$ from Lemma 36 finishes the proof. □

**Corollary 39.** *For any psd matrix $\mathbf{Q} \succeq 0$, we have:*

$$\begin{aligned}\left\|\mathbf{A}^{n+1-j}\mathcal{B}^j\mathbf{Q}\right\| &\le \frac{12\sqrt{2}(n+1-j)\kappa}{\sqrt{1-\alpha^2}}\alpha^{\frac{n-j}{2}}\left(1-\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right)^j\|\mathbf{Q}\|_2 \\ &\le \frac{12\sqrt{2}(n+1-j)\kappa}{\sqrt{1-\alpha^2}}\alpha^{\frac{n-j}{2}}\exp\left(\frac{-jc_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right)\|\mathbf{Q}\|_2\,.\end{aligned}$$

*Proof.* This corollary follows directly from Lemmas 34 and 37 and using the fact that $1-x \leq e^{-x}$ □

The following lemma bounds the total error of $\bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}}$.

**Lemma 40.**

$$\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n}^{bias} \otimes \bar{\boldsymbol{\theta}}_{t,n}^{bias}\right] \rangle \leq C \cdot \frac{(\kappa\widetilde{\kappa})^{9/4} d\kappa}{(n-t)^2} \cdot \exp\left( -(t+1)\frac{c_2 c_3 \sqrt{2c_1 - c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \left(P(\mathbf{x}_0) - P(\mathbf{x}^*)\right)$$
$$+ C \cdot (\kappa\widetilde{\kappa})^{5/4} d\kappa \cdot \exp\left(\frac{-nc_2 c_3 \sqrt{2c_1 - c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \left(P(\mathbf{x}_0) - P(\mathbf{x}^*)\right)$$

*Where, $C$ is a universal constant.*

*Proof.* Lemma 11 tells us that

$$\mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}} \otimes \bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}}\right] = \frac{1}{(n-t)^2}\Big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\Big)\cdot$$
$$(\mathcal{I} - \mathcal{B})^{-1}(\mathcal{B}^{t+1} - \mathcal{B}^{n+1})\left(\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0\right)$$
$$- \frac{1}{(n-t)^2}\sum_{j=t+1}^{n}\Big((\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}}^{n+1-j} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j}\Big)\mathcal{B}^j\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0. \tag{B.24}$$

We now use lemmas in this section to bound inner product of the two terms in the above expression with $\begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}$, i.e. we seek to bound,

$$\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}} \otimes \bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}}\right] \rangle$$
$$= \langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \Big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\Big)(\mathcal{I} - \mathcal{B})^{-1}(\mathcal{B}^{t+1} - \mathcal{B}^{n+1})\left(\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0\right)/(n-t)^2\rangle$$
$$+ \langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, -\frac{1}{(n-t)^2}\sum_{j=t+1}^{n}\Big((\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}}^{n+1-j} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j}\Big)\mathcal{B}^j\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0\rangle \tag{B.25}$$

For the first term of equation (B.25), we have

$$\begin{aligned}
&\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \Big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\Big)(\mathcal{I} - \mathcal{B})^{-1}(\mathcal{B}^{t+1} - \mathcal{B}^{n+1})\left(\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0\right)\rangle \\
&= \Big\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \Big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\Big)\left(\mathcal{I} - \mathcal{A}_{\mathcal{L}}\mathcal{A}_{\mathcal{R}}^{\top}\right)^{-1}\left(\mathcal{I} - \mathcal{A}_{\mathcal{L}}\mathcal{A}_{\mathcal{R}}^{\top}\right) \\
&\qquad\qquad (\mathcal{I} - \mathcal{B})^{-1}(\mathcal{B}^{t+1} - \mathcal{B}^{n+1})\left(\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0\right)\Big\rangle \\
&= \Big\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}(\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\left(\mathcal{I} - \mathcal{A}_{\mathcal{L}}\mathcal{A}_{\mathcal{R}}^{\top}\right)(\mathcal{I} - \mathcal{B})^{-1}(\mathcal{B}^{t+1} - \mathcal{B}^{n+1})\left(\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0\right)\Big\rangle \\
&\qquad\qquad \text{(using Lemma 35)} \\
&= \langle (\mathbf{I} - \mathbf{A}^{\top})^{-1}\begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}(\mathbf{I} - \mathbf{A})^{-1}, (\mathcal{I} - \mathcal{D})\left(\mathcal{I} - \mathcal{B}\right)^{-1}(\mathcal{B}^{t+1} - \mathcal{B}^{n+1})\left(\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0\right)\rangle \\
&\le \frac{1}{(q - c\delta)^2}\frac{4\kappa}{\sqrt{1-\alpha^2}}\exp\left(-(t+1)c_2c_3\sqrt{2c_1 - c_1^2}/\sqrt{\kappa\widetilde{\kappa}}\right)\|\boldsymbol{\theta}_0\|^2 \\
&\qquad\qquad \langle\Big(\otimes_2 \begin{bmatrix} -(c\mathbf{I} - q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I} - \delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Big), \mathbf{I} + 2\sqrt{\kappa\widetilde{\kappa}}(R^2/\sigma^2)\widehat{\mathbf{\Sigma}}\rangle.
\end{aligned}$$

The two terms above can be bounded as

$$\langle\Big(\otimes_2 \begin{bmatrix} -(c\mathbf{I} - q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I} - \delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Big), \mathbf{I}\rangle \le 7 \cdot \operatorname{Tr}\left(\mathbf{H}^{-1}\right) \le \frac{7d}{\mu} \text{ and,}$$

$$2\sqrt{\kappa\widetilde{\kappa}}(R^2/\sigma^2)\langle\Big(\otimes_2 \begin{bmatrix} -(c\mathbf{I} - q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I} - \delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Big), \widehat{\mathbf{\Sigma}}\rangle = 2\sqrt{\kappa\widetilde{\kappa}}R^2(q - c\delta)^2 d.$$

Combining the above and noting the fact that $2\sqrt{\kappa\widetilde{\kappa}}R^2(q-c\delta)^2 d < \frac{7d}{\mu}$, we have

$$\begin{aligned}
&\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \Big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\Big)(\mathcal{I} - \mathcal{B})^{-1}(\mathcal{B}^{t+1} - \mathcal{B}^{n+1})\left(\boldsymbol{\theta}_0 \otimes \boldsymbol{\theta}_0\right)\rangle \\
&\qquad \le \frac{56\kappa d}{\sqrt{1-\alpha^2}} \cdot \frac{\|\boldsymbol{\theta}_0\|^2}{\mu\left(q - c\delta\right)^2} \cdot \exp\left(-(t+1)c_2c_3\sqrt{2c_1 - c_1^2}/\sqrt{\kappa\widetilde{\kappa}}\right). \qquad \text{(B.26)}
\end{aligned}$$

We now note the following facts:

$$\frac{1}{1-\alpha} = \frac{c_2\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}+c_2\sqrt{2c_1-c_1^2}} \le \frac{2}{\sqrt{c_1c_4}}\cdot\sqrt{\kappa\widetilde{\kappa}}$$
$$\frac{1}{q-c\delta} \le \frac{1}{\gamma(1-\alpha)} \le \frac{\mu}{(1-\alpha)^2} \le \frac{4\widetilde{\kappa}}{c_4\delta}$$

This implies, equation (B.26) can be bounded as:

$$\begin{aligned}
&\langle \begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix}, \Big(\mathcal{I}+(\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}}+(\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\Big)(\mathcal{I}-\mathcal{B})^{-1}(\mathcal{B}^{t+1}-\mathcal{B}^{n+1})\left(\boldsymbol{\theta}_0\otimes\boldsymbol{\theta}_0\right)\rangle\\
&\quad\le \frac{1792}{(c_1c_4)^{5/4}}\cdot\frac{(\kappa\widetilde{\kappa})^{9/4}d}{\delta c_4}\cdot\exp\Big(-(t+1)\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\Big)\left\|\boldsymbol{\theta}_0\right\|^2\\
&\quad\le \frac{1792}{(c_1c_4)^{5/4}}\cdot\frac{(\kappa\widetilde{\kappa})^{9/4}d\kappa}{c_1c_4}\cdot\exp\Big(-(t+1)\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\Big)\mu\left\|\boldsymbol{\theta}_0\right\|^2\\
&\quad\le \frac{3584}{(c_1c_4)^{5/4}}\cdot\frac{(\kappa\widetilde{\kappa})^{9/4}d\kappa}{c_1c_4}\cdot\exp\Big(-(t+1)\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\Big)\cdot\big(P(\mathbf{x}_0)-P(\mathbf{x}^*)\big)\\
&\quad\le C\cdot(\kappa\widetilde{\kappa})^{9/4}d\kappa\cdot\exp\Big(-(t+1)\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\Big)\cdot\big(P(\mathbf{x}_0)-P(\mathbf{x}^*)\big). \qquad \text{(B.27)}
\end{aligned}$$

Where, $C$ is a universal constant.

Consider now a term in the summation in the second term of (B.25).

$$\begin{aligned}
&\langle \begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix}, \Big((\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}}^{n+1-j}+(\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{A}_{\mathcal{R}}^{\top})^{n+1-j}\Big)\mathcal{B}^j\left(\boldsymbol{\theta}_0\otimes\boldsymbol{\theta}_0\right)\rangle\\
&= \langle (\mathbf{I}-\mathbf{A}^{\top})^{-1}\begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix}, \mathbf{A}^{n+1-j}\mathcal{B}^j\left(\boldsymbol{\theta}_0\otimes\boldsymbol{\theta}_0\right)\rangle\\
&+\langle \begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix}(\mathbf{I}-\mathbf{A})^{-1}, \Big(\mathcal{B}^j\left(\boldsymbol{\theta}_0\otimes\boldsymbol{\theta}_0\right)\Big)(\mathbf{A}^{\top})^{n+1-j}\rangle\\
&\le 4d\left\|(\mathbf{I}-\mathbf{A}^{\top})^{-1}\begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix}\right\|\left\|\mathbf{A}^{n+1-j}\mathcal{B}^j\left(\boldsymbol{\theta}_0\otimes\boldsymbol{\theta}_0\right)\right\|\\
&\le \frac{4d}{q-c\delta}\left\|\begin{bmatrix}-(c\mathbf{I}-q\mathbf{H}) & 0\\ (\mathbf{I}-\delta\mathbf{H}) & 0\end{bmatrix}\right\|\cdot\frac{12\sqrt{2}(n+1-j)\kappa}{\sqrt{1-\alpha^2}}\alpha^{\frac{n-j}{2}}\exp\left(\frac{-jc_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right)\left\|\boldsymbol{\theta}_0\right\|^2
\end{aligned}$$

(Lemma 32 and Corollary 39)

$$
\begin{aligned}
&\le \frac{672(n-t)d\kappa}{(q-c\delta)\sqrt{1-\alpha^2}} \cdot \exp\left(\frac{-nc_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \|\boldsymbol{\theta}_0\|^2 \\
&\le \frac{5376}{(c_1c_4)^{1/4}} \frac{(\kappa\widetilde{\kappa})^{5/4}d}{\delta c_4}(n-t)\exp\left(\frac{-nc_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \|\boldsymbol{\theta}_0\|^2 \\
&\le \frac{5376}{(c_1c_4)^{1/4}} \frac{(\kappa\widetilde{\kappa})^{5/4}d\kappa}{c_1c_4}(n-t)\exp\left(\frac{-nc_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \mu\|\boldsymbol{\theta}_0\|^2 \\
&\le \frac{10752}{(c_1c_4)^{1/4}} \frac{(\kappa\widetilde{\kappa})^{5/4}d\kappa}{c_1c_4}(n-t)\exp\left(\frac{-nc_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \big(P(\mathbf{x}_0)-P(\mathbf{x}^*)\big) \\
&\le C\cdot(\kappa\widetilde{\kappa})^{5/4}d\kappa\cdot(n-t)\exp\left(\frac{-nc_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \big(P(\mathbf{x}_0)-P(\mathbf{x}^*)\big). \qquad \text{(B.28)}
\end{aligned}
$$

Where, $C$ is a universal constant. Plugging (B.27) and (B.28) into (B.25), we obtain

$$
\begin{aligned}
&\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}} \otimes \bar{\boldsymbol{\theta}}_{t,n}^{\text{bias}}\right]\rangle \\
&\le C\cdot\frac{(\kappa\widetilde{\kappa})^{9/4}d\kappa}{(n-t)^2}\cdot\exp\left(-(t+1)\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right)\cdot\big(P(\mathbf{x}_0)-P(\mathbf{x}^*)\big) \\
&\qquad + C\cdot(\kappa\widetilde{\kappa})^{5/4}d\kappa\cdot\exp\left(\frac{-nc_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right)\cdot\big(P(\mathbf{x}_0)-P(\mathbf{x}^*)\big)
\end{aligned}
$$

This proves the lemma. □

## B.5 Lemmas and proofs for Bounding variance error

Before we prove Lemma 14, we recall old notation and introduce new notations that will be employed in these proofs.

### *B.5.1 Notations*

We begin with by recalling that we track $\boldsymbol{\theta}_k = \begin{bmatrix} \mathbf{x}_k - \mathbf{x}^* \\ \mathbf{y}_k - \mathbf{x}^* \end{bmatrix}$. Given $\boldsymbol{\theta}_k$, we recall the recursion governing the evolution of $\boldsymbol{\theta}_k$:

$$\begin{aligned} \boldsymbol{\theta}_{k+1} &= \begin{bmatrix} 0 & \mathbf{I} - \delta\widehat{\mathbf{H}}_{k+1} \\ -c \cdot \mathbf{I} & (1+c)\mathbf{I} - q \cdot \widehat{\mathbf{H}}_{k+1} \end{bmatrix} \boldsymbol{\theta}_k + \begin{bmatrix} \delta \cdot \epsilon_{k+1}\mathbf{a}_{k+1} \\ q \cdot \epsilon_{k+1}\mathbf{a}_{k+1} \end{bmatrix} \\ &= \widehat{\mathbf{A}}_{k+1}\boldsymbol{\theta}_k + \boldsymbol{\zeta}_{k+1} \end{aligned} \tag{B.29}$$

where, recall, $c = \alpha(1-\beta)$, $q = \alpha\delta + (1-\alpha)\gamma$, and $\widehat{\mathbf{H}}_{k+1} = \mathbf{a}_{k+1}\mathbf{a}_{k+1}^\top$. Furthermore, we recall the following definitions, which will be heavily used in the following proofs:

$$\begin{aligned} \mathbf{A} &= \mathbb{E}\left[\widehat{\mathbf{A}}_{k+1}|\mathcal{F}_k\right] \\ \mathcal{B} &= \mathbb{E}\left[\widehat{\mathbf{A}}_{k+1} \otimes \widehat{\mathbf{A}}_{k+1}|\mathcal{F}_k\right] \\ \widehat{\boldsymbol{\Sigma}} &= \mathbb{E}\left[\boldsymbol{\zeta}_{k+1} \otimes \boldsymbol{\zeta}_{k+1}|\mathcal{F}_k\right] = \begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \boldsymbol{\Sigma} \preceq \sigma^2 \cdot \begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{H} \end{aligned}$$

We recall:

$$\begin{aligned} \mathcal{R} &= \mathbb{E}\left[(\mathbf{A} - \widehat{\mathbf{A}}_{k+1}) \otimes (\mathbf{A} - \widehat{\mathbf{A}}_{k+1})|\mathcal{F}_k\right] \\ \mathcal{D} &= \mathbf{A} \otimes \mathbf{A} \end{aligned}$$

And the operators $\mathcal{B}, \mathcal{D}, \mathcal{R}$ being related by:

$$\mathcal{B} = \mathcal{D} + \mathcal{R}$$

Furthermore, in order to compute the steady state distribution with the fourth moment quantities in the mix, we need to rely on the following re-parameterization of the update matrix $\widehat{\mathbf{A}}$:

$$\widehat{\mathbf{A}} = \begin{bmatrix} 0 & \mathbf{I} - \delta\widehat{\mathbf{H}} \\ -c \cdot \mathbf{I} & (1+c) \cdot \mathbf{I} - q \cdot \widehat{\mathbf{H}} \end{bmatrix}$$

$$\begin{aligned}
&= \begin{bmatrix} 0 & \mathbf{I} \\ -c\cdot\mathbf{I} & (1+c)\cdot\mathbf{I} \end{bmatrix} + \begin{bmatrix} 0 & -\delta\cdot\widehat{\mathbf{H}} \\ 0 & -q\cdot\widehat{\mathbf{H}} \end{bmatrix} \\
&\overset{\text{def}}{=} \mathbf{V}_1 + \hat{\mathbf{V}}_2
\end{aligned}$$

This implies in particular:

$$\begin{aligned}
\widehat{\mathbf{A}}\otimes\widehat{\mathbf{A}} &= (\mathbf{V}_1+\hat{\mathbf{V}}_2)\otimes(\mathbf{V}_1+\hat{\mathbf{V}}_2) \\
&= \mathbf{V}_1\otimes\mathbf{V}_1 + \mathbf{V}_1\otimes\hat{\mathbf{V}}_2 + \hat{\mathbf{V}}_2\otimes\mathbf{V}_1 + \hat{\mathbf{V}}_2\otimes\hat{\mathbf{V}}_2
\end{aligned}$$

Note in particular, the fourth moment part resides in the operator $\hat{\mathbf{V}}_2\otimes\hat{\mathbf{V}}_2$. Terms such as $\mathbf{V}_1\otimes\mathbf{V}_1$ are deterministic, or terms such as $\mathbf{V}_1\otimes\hat{\mathbf{V}}_2$ or $\hat{\mathbf{V}}_2\otimes\mathbf{V}_1$ contain second moment quantities. Furthermore, note that the operator $\mathcal{B} = \mathbb{E}\left[\widehat{\mathbf{A}}\otimes\widehat{\mathbf{A}}\right]$ where the expectation is taken with respect to a single random draw from the distribution $\mathcal{D}$.

Considering the expectation of $\widehat{\mathbf{A}}\otimes\widehat{\mathbf{A}}$ with respect to a single draw from the distribution $\mathcal{D}$, we have:

$$\begin{aligned}
\mathcal{B} = \mathbb{E}\left[\widehat{\mathbf{A}}\otimes\widehat{\mathbf{A}}\right] &= \mathbf{V}_1\otimes\mathbf{V}_1 + \mathbb{E}\left[\mathbf{V}_1\otimes\hat{\mathbf{V}}_2\right] + \mathbb{E}\left[\hat{\mathbf{V}}_2\otimes\mathbf{V}_1\right] + \mathbb{E}\left[\hat{\mathbf{V}}_2\otimes\hat{\mathbf{V}}_2\right] \\
&= \mathbf{V}_1\otimes\mathbf{V}_1 + \mathbf{V}_1\otimes\mathbf{V}_2 + \mathbf{V}_2\otimes\mathbf{V}_1 + \mathbb{E}\left[\hat{\mathbf{V}}_2\otimes\hat{\mathbf{V}}_2\right],
\end{aligned}$$

where $\mathbf{V}_2 \overset{\text{def}}{=} \mathbb{E}\left[\hat{\mathbf{V}}_2\right] = \begin{bmatrix} 0 & -\delta\cdot\mathbf{H} \\ 0 & -q\cdot\mathbf{H} \end{bmatrix}$.

Finally, we let nr and dr to denote the numerator and denominator respectively.

### *B.5.2 An exact expression for the stationary distribution*

Note that a key term appearing in the expression for covariance of the variance equation (B.15) is $(\mathcal{I}-\mathcal{B})^{-1}\widehat{\boldsymbol{\Sigma}}$. This is in fact nothing but the covariance of the error when we run accelerated SGD forever starting at $\mathbf{x}^*$ (i.e., at steady state). This can be seen by considering the base variance recursion using equation (B.29):

$$\boldsymbol{\theta}_k = \widehat{\mathbf{A}}_k\boldsymbol{\theta}_{k-1} + \boldsymbol{\zeta}_k$$

$$\begin{aligned}
\implies \mathbf{\Phi}_k &\overset{\text{def}}{=} \mathbb{E}\left[\boldsymbol{\theta}_k \otimes \boldsymbol{\theta}_k\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\widehat{\mathbf{A}}_k \boldsymbol{\theta}_{k-1} \otimes \boldsymbol{\theta}_{k-1} \widehat{\mathbf{A}}_k^\top + \boldsymbol{\zeta}_k \otimes \boldsymbol{\zeta}_k\right) | \mathcal{F}_{k-1}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\widehat{\mathbf{A}}_k \boldsymbol{\theta}_{k-1} \otimes \boldsymbol{\theta}_{k-1} \widehat{\mathbf{A}}_k^\top\right) | \mathcal{F}_{k-1}\right]\right] + \widehat{\mathbf{\Sigma}} \\
&= \mathcal{B} \cdot \mathbb{E}\left[\boldsymbol{\theta}_{k-1} \otimes \boldsymbol{\theta}_{k-1}\right] + \widehat{\mathbf{\Sigma}} \\
&= \mathcal{B} \cdot \mathbf{\Phi}_{k-1} + \widehat{\mathbf{\Sigma}}
\end{aligned}$$

This recursion on the covariance operator $\mathbf{\Phi}_k$ can be unrolled until the start i.e. $k = 0$ to yield:

$$\begin{aligned}
\mathbf{\Phi}_k &= \mathcal{B}^k \mathbf{\Phi}_0 + \sum_{l=0}^{k-1} \mathcal{B}^l \cdot \widehat{\mathbf{\Sigma}} \\
&= (\mathcal{I} - \mathcal{B})^{-1} (\mathcal{I} - \mathcal{B}^k) \widehat{\mathbf{\Sigma}} \qquad (\because\ \mathbf{\Phi}_0 = 0) \\
\implies \mathbf{\Phi}_\infty &= \lim_{k \to \infty} \mathbf{\Phi}_k = (\mathcal{I} - \mathcal{B})^{-1} \widehat{\mathbf{\Sigma}} \qquad \text{(B.30)}
\end{aligned}$$

*B.5.3 Computing the steady state distribution*

We now proceed to compute the stationary distribution. Recall that

$$\begin{aligned}
\mathcal{B} &= \mathbf{V}_1 \otimes \mathbf{V}_1 + \mathbf{V}_1 \otimes \mathbf{V}_2 + \mathbf{V}_2 \otimes \mathbf{V}_1 + \mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right] \\
\implies \mathcal{I} - \mathcal{B} &= \left(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\right) - \mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right]
\end{aligned}$$

Where the expectation is over a single sample drawn from the distribution $\mathcal{D}$. This implies in particular,

$$\begin{aligned}
(\mathcal{I} - \mathcal{B})^{-1} &= \left(\left(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\right) - \mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right]\right)^{-1} \\
&= \sum_{k=0}^{\infty} \left(\left(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\right)^{-1} \mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right]\right)^k \\
&\qquad \cdot \left(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\right)^{-1} \qquad \text{(B.31)}
\end{aligned}$$

Since $\widehat{\mathbf{\Sigma}} \preceq \sigma^2 \cdot \begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{H}$, and $(\mathcal{I} - \mathcal{B})^{-1}$ is a PSD operator, the steady state distribution $\mathbf{\Phi}_\infty$ is bounded by:

$$\begin{aligned}\mathbf{\Phi}_\infty = (\mathcal{I} - \mathcal{B})^{-1}\widehat{\mathbf{\Sigma}} &\preceq \sigma^2(\mathcal{I} - \mathcal{B})^{-1}\left(\begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{H}\right) \\ &= \sigma^2 \sum_{k=0}^{\infty}\left(\big(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\big)^{-1}\mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right]\right)^k \cdot \\ &\qquad \big(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\big)^{-1}\left(\begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{H}\right). \qquad \text{(B.32)}\end{aligned}$$

Note that the Taylor expansion above is guaranteed to be correct if the right hand side is finite. We will understand bounds on the steady state distribution by splitting the analysis into the following parts:

- Obtain $\mathbf{U} \overset{\text{def}}{=} \big(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\big)^{-1}\left(\begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{H}\right)$ (in section B.5.3).
- Obtain bounds on $\mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right]\mathbf{U}$ (in section B.5.3)
- Combine the above to obtain bounds on $\mathbf{\Phi}_\infty$ (Lemma 14).

Before deriving these bounds, we will present some reasoning behind the validity of the upper bounds that we derive on the stationary distribution $\mathbf{\Phi}_\infty$:

$$\begin{aligned}\mathbf{\Phi}_\infty &= (\mathcal{I} - \mathcal{B})^{-1}\widehat{\mathbf{\Sigma}} \\ &\preceq \sigma^2 \sum_{k=0}^{\infty}\left(\big(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\big)^{-1}\mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right]\right)^k \mathbf{U} \quad (***) \\ &= \sigma^2\mathbf{U} + \sigma^2 \sum_{k=1}^{\infty}\left(\big(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\big)^{-1}\mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right]\right)^k \mathbf{U} \\ &= \sigma^2\mathbf{U} + \sigma^2 \sum_{k=0}^{\infty}\left(\big(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\big)^{-1}\mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right]\right)^k\end{aligned}$$

$$\cdot \big(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\big)^{-1} \mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right] \mathbf{U}$$

$$= \sigma^2 \mathbf{U} + \sigma^2 (\mathcal{I} - \mathcal{B})^{-1} \cdot \mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right] \mathbf{U} \qquad \text{(using equation (B.31))}, \tag{B.33}$$

with $(***)$ following through using equation (B.32) and through the definition of $\mathbf{U}$. Now, with this in place, we clearly see that since $(\mathcal{I} - \mathcal{B})^{-1}$ and $\mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right]$ are PSD operators, we can upper bound right hand side to create valid PSD upper bounds on $\mathbf{\Phi}_\infty$. In particular, in section B.5.3, we derive with equality what $\mathbf{U}$ is, and follow that up with computation of an upper bound on $\mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right] \mathbf{U}$ in section B.5.3. Combining this will enable us to present a valid PSD upper bound on $\mathbf{\Phi}_\infty$ owing to equation (B.33).

*Understanding the second moment effects*

This part of the proof deals with deriving the solution to:

$$\mathbf{U} = \big(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\big)^{-1} \left( \begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{H} \right)$$

This is equivalent to solving the (linear) equation:

$$\big(\mathcal{I} - \mathbf{V}_1 \otimes \mathbf{V}_1 - \mathbf{V}_1 \otimes \mathbf{V}_2 - \mathbf{V}_2 \otimes \mathbf{V}_1\big) \cdot \mathbf{U} = \left( \begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{H} \right)$$

$$\implies \mathbf{U} - \mathbf{V}_1 \mathbf{U} \mathbf{V}_1^\top - \mathbf{V}_1 \mathbf{U} \mathbf{V}_2^\top - \mathbf{V}_2 \mathbf{U} \mathbf{V}_1^\top = \left( \begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{H} \right) \tag{B.34}$$

Note that all the known matrices above i.e., $\mathbf{V}_1, \mathbf{V}_2$ and $\mathbf{H}$ are all diagonalizable with respect to $\mathbf{H}$, and thus, the solution of this system can be computed in each of the eigenspaces $(\lambda_j, \mathbf{u}_j)$ of $\mathbf{H}$. This implies, in reality, we deal with matrices $\mathbf{U}^{(j)}$, one corresponding to each eigenspace. However, for this section, we will neglect the superscript on $\mathbf{U}$, since it is clear from context for the purpose of this section.

$$\mathbf{V}_1 \mathbf{U} \mathbf{V}_1^\top = \begin{bmatrix} 0 & 1 \\ -c & 1+c \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ u_{12} & u_{22} \end{bmatrix} \begin{bmatrix} 0 & -c \\ 1 & 1+c \end{bmatrix}$$

$$= \begin{bmatrix} u_{22} & -cu_{12} + (1+c)u_{22} \\ -cu_{12} + (1+c)u_{22} & c^2u_{11} - 2c(1+c)u_{12} + (1+c)^2u_{22} \end{bmatrix}$$

Next,

$$\begin{aligned} \mathbf{V}_1\mathbf{U}\mathbf{V}_2^\top &= \begin{bmatrix} 0 & 1 \\ -c & 1+c \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ u_{12} & u_{22} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ -\delta & -q \end{bmatrix} \lambda_j \\ &= \begin{bmatrix} u_{12} & u_{22} \\ -cu_{11} + (1+c)u_{12} & -cu_{12} + (1+c)u_{22} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ -\delta & -q \end{bmatrix} \lambda_j \\ &= \begin{bmatrix} -\delta u_{22} & -qu_{22} \\ -\delta(-cu_{12} + (1+c)u_{22}) & -q(-cu_{12} + (1+c)u_{22}) \end{bmatrix} \lambda_j \end{aligned}$$

It follows that:

$$\begin{aligned} \mathbf{V}_2\mathbf{U}\mathbf{V}_1^\top &= (\mathbf{V}_1\mathbf{U}\mathbf{V}_2^\top)^\top \\ &= \begin{bmatrix} -\delta u_{22} & -\delta(-cu_{12} + (1+c)u_{22}) \\ -qu_{22} & -q(-cu_{12} + (1+c)u_{22}) \end{bmatrix} \lambda_j \end{aligned}$$

Given all these computations, comparing the $(1,1)$ term on both sides of equation (B.34), we get:

$$\begin{aligned} u_{11} - u_{22} + 2\delta\lambda_j u_{22} &= \delta^2\lambda_j \\ u_{11} &= u_{22}(1 - 2\delta\lambda_j) + \delta^2\lambda_j \end{aligned} \tag{B.35}$$

Next, comparing $(1,2)$ term on both sides of equation (B.34), we get:

$$\begin{aligned} &u_{12} - (-cu_{12} + (1+c)u_{22}) + q\lambda_j u_{22} + \delta\lambda_j(-cu_{12} + (1+c)u_{22}) = \delta\ q\lambda_j \\ &u_{12} - (1 - \delta\lambda_j)(-cu_{12} + (1+c)u_{22}) + q\lambda_j u_{22} = \delta\ q\lambda_j \\ &(1 + c(1 - \delta\lambda_j)) \cdot u_{12} + (q\lambda_j - (1+c)(1 - \delta\lambda_j)) \cdot u_{22} = \delta\ q\lambda_j \end{aligned} \tag{B.36}$$

Finally, comparing the $(2,2)$ term on both sides of equation (B.34), we get:

$$u_{22} - (c^2u_{11} - 2c(1+c)u_{12} + (1+c)^2u_{22}) + 2q\lambda_j(-cu_{12} + (1+c)u_{22}) = q^2\lambda_j$$

$$\implies -c^2u_{11} + (2c(1+c) - 2cq\lambda_j)u_{12} + (1-(1+c)^2 + 2(1+c)q\lambda_j)u_{22} = q^2\lambda_j \quad \text{(from eq (B.35))}$$

$$\implies -c^2(u_{22}(1-2\delta\lambda_j) + \delta^2\lambda_j) + (2c(1+c) - 2cq\lambda_j)u_{12} + (1-(1+c)^2 + 2(1+c)q\lambda_j)u_{22} = q^2\lambda_j$$

$$\implies (2c(1+c) - 2cq\lambda_j)u_{12} + (1-(1+c)^2 - c^2(1-2\delta\lambda_j) + 2(1+c)q\lambda_j)u_{22} = (q^2 + c^2\delta^2)\lambda_j$$

$$\implies 2c((1+c) - q\lambda_j)u_{12} + 2((1+c)(q\lambda_j - c) + \delta\lambda_j c^2)u_{22} = (q^2 + c^2\delta^2)\lambda_j \tag{B.37}$$

Now, we note that equations (B.36), (B.37) are linear systems in two variables $u_{12}$ and $u_{22}$. Denoting the system in the following manner,

$$a_{11}u_{12} + a_{12}u_{22} = b_1$$

$$a_{21}u_{12} + a_{22}u_{22} = b_2$$

For analyzing the variance error, we require $u_{22}, u_{12}$:

$$u_{22} = \frac{b_1a_{21} - b_2a_{11}}{a_{12}a_{21} - a_{11}a_{22}}, \; u_{12} = \frac{b_1a_{22} - b_2a_{12}}{a_{11}a_{22} - a_{12}a_{21}}$$

Substituting the values from equations (B.36) and (B.37), we get:

$$u_{22} = \frac{2cq\delta\Big(1+c-q\lambda_j\Big) - (q^2+c^2\delta^2)\Big(1+c(1-\delta\lambda_j)\Big)}{\mathrm{dr}(u_{22})} \cdot \lambda_j \tag{B.38}$$

$$u_{12} = \frac{2q\delta\Big((1+c)(q\lambda_j - c) + \delta\lambda_j c^2\Big) - (q^2+c^2\delta^2)\Big(\lambda_j q - (1+c)(1-\delta\lambda_j)\Big)}{\mathrm{dr}(u_{12})} \cdot \lambda_j \tag{B.39}$$

$$\begin{aligned}\mathrm{dr}(u_{22}) = 2c&\Big(\big(1+c-q\lambda_j\big)\cdot\big(\lambda_j q - (1+c)(1-\delta\lambda_j)\big)\Big)\\ &- 2\cdot\Big(\big(1+c-c\delta\lambda_j\big)\cdot\big((1+c)(q\lambda_j - c) + \delta\lambda_j c^2\big)\Big)\end{aligned} \tag{B.40}$$

$$\begin{aligned}\mathrm{dr}(u_{12}) = 2&\Big(\big(1+c-c\delta\lambda_j\big)\cdot\big((1+c)(q\lambda_j - c) + \delta\lambda_j c^2\big)\Big)\\ &- 2c\Big(\big(1+c-q\lambda_j\big)\cdot\big(\lambda_j q - (1+c)(1-\delta\lambda_j)\big)\Big)\end{aligned} \tag{B.41}$$

**Denominator of $u_{22}$**: Let us consider the denominator of $u_{22}$ (from equation (B.38)) to write it in a concise manner.

$$\mathrm{dr}(u_{22}) = 2\Big( \big(1+c-q\lambda_j\big)\cdot k_1 \ - \ \big(1+c-c\delta\lambda_j\big)\cdot k_2\Big)$$

with

$$\begin{aligned}
k_1 &= c\cdot\big(\lambda_j q-(1+c)(1-\delta\lambda_j)\big)\\
&= \big(c\lambda_j q-(c+c^2)(1-\delta\lambda_j)\big)\\
&= \big(cq\lambda_j-c-c^2+c\delta\lambda_j+c^2\delta\lambda_j\big)\\
k_2 &= \big((1+c)(q\lambda_j-c)+\delta\lambda_j c^2\big)\\
&= \big(q\lambda_j-c+cq\lambda_j-c^2+\delta\lambda_j c^2\big)
\end{aligned}$$

Plugging in expressions for $q=\alpha\delta+(1-\alpha)\gamma$ and $c=\alpha(1-\beta)$, in $\mathrm{dr}(u_{22})$ we get:

$$\mathrm{dr}(u_{22}) = 2\cdot\Bigg(\ \big(1+c-\alpha\delta\lambda_j\big)(k_1-k_2)-\lambda_j\cdot\big((1-\alpha)\gamma k_1+\alpha\beta\delta k_2\big)\ \Bigg) \tag{B.42}$$

Next, considering $k_1-k_2$, we have:

$$\begin{aligned}
k_1-k_2 &= c\lambda_j q-c-c^2+c\delta\lambda_j+c^2\delta\lambda_j-q\lambda_j+c-cq\lambda_j+c^2-c^2\delta\lambda_j\\
&= (c\delta-q)\lambda_j\\
&= -(\alpha\beta\delta+\gamma(1-\alpha))\lambda_j
\end{aligned} \tag{B.43}$$

Next, considering $\gamma(1-\alpha)k_1+\alpha\beta\delta\ k_2$, we have:

$$\begin{aligned}
&\gamma(1-\alpha)k_1+\alpha\beta\delta\ k_2\\
&= \gamma(1-\alpha)(c\lambda_j q-c-c^2+c^2\delta\lambda_j+c\delta\lambda_j)\\
&+\alpha\beta\delta(c\lambda_j q-c-c^2+c^2\delta\lambda_j+q\lambda_j)\\
&= (\alpha\beta\delta+(1-\alpha)\gamma)(c\lambda_j q-c-c^2+c^2\delta\lambda_j)+\lambda_j\delta(c\gamma(1-\alpha)+\alpha\beta q)
\end{aligned}$$

Consider $c\gamma(1-\alpha)+\alpha\beta q$:

$$c\gamma(1-\alpha)+\alpha\beta q = \alpha(1-\beta)\gamma(1-\alpha)+\alpha\beta(\alpha\delta+(1-\alpha)\gamma)$$

$$= \alpha(1-\beta)\gamma(1-\alpha) + \alpha\beta\gamma(1-\alpha) + \alpha^2\beta\delta$$
$$= \alpha\gamma(1-\alpha) + \alpha^2\beta\delta$$
$$= \alpha(\alpha\beta\delta + (1-\alpha)\gamma)$$

Re-substituting this in the expression for $\gamma(1-\alpha)k_1 + \alpha\beta\delta k_2$, we have:

$$\begin{aligned}\gamma(1-\alpha)k_1 + \alpha\beta\delta\ k_2 &= (\alpha\beta\delta + (1-\alpha)\gamma)(c\lambda_j q - c - c^2 + c^2\delta\lambda_j) + \lambda_j\delta(c\gamma(1-\alpha) + \alpha\beta q)\\ &= (\alpha\beta\delta + (1-\alpha)\gamma)(c\lambda_j q - c - c^2 + c^2\delta\lambda_j) + \alpha\lambda_j\delta(\alpha\beta\delta + (1-\alpha)\gamma)\\ &= (\alpha\beta\delta + (1-\alpha)\gamma)(c\lambda_j q - c - c^2 + c^2\delta\lambda_j + \alpha\lambda_j\delta) \qquad \text{(B.44)}\end{aligned}$$

Substituting equations (B.43), (B.44) into equation (B.42), we have:

$$\begin{aligned}\mathrm{dr}(u_{22}) &= -2\lambda_j(\alpha\beta\delta + \gamma(1-\alpha)) \cdot (1 + c - \alpha\delta\lambda_j + c\lambda_j q - c - c^2 + c^2\delta\lambda_j + \alpha\delta\lambda_j)\\ &= -2\lambda_j(\alpha\beta\delta + \gamma(1-\alpha)) \cdot (1 - c^2 + c\lambda_j(q + c\delta)) \qquad \text{(B.45)}\end{aligned}$$

We note that the denominator of $u_{12}$ (in equation (B.39)) is just the negative of the denominator of $u_{22}$ as represented in equation (B.45).

**Numerator of $u_{22}$**: We begin by writing out the numerator of $u_{22}$ (from equation (B.38)):

$$\begin{aligned}\mathrm{nr}(u_{22}) &= \lambda_j \cdot \Big(2cq\delta\big(1 + c - q\lambda_j\big) - (q^2 + c^2\delta^2)\big(1 + c(1 - \delta\lambda_j)\big)\Big)\\ &= \lambda_j \cdot \Big(2cq\delta\big(1 + c - \alpha\delta\lambda_j - \gamma(1-\alpha)\lambda_j\big) - (q^2 + c^2\delta^2)\big(1 + c - \alpha\delta\lambda_j + \alpha\beta\delta\lambda_j\big)\Big)\\ &= \lambda_j \cdot \Big( -(1 + c - \alpha\delta\lambda_j)(q - c\delta)^2 - \lambda_j \cdot \big(2cq\delta\gamma(1-\alpha) + (q^2 + (c\delta)^2)\alpha\beta\delta\big)\Big)\end{aligned} \qquad \text{(B.46)}$$

We now consider $2cq\delta\gamma(1-\alpha) + (q^2 + (c\delta)^2)\alpha\beta\delta$:

$$\begin{aligned}&2cq\delta\gamma(1-\alpha) + (q^2 + (c\delta)^2)\alpha\beta\delta\\ &= 2cq\delta \cdot (\gamma(1-\alpha) + \alpha\beta\delta) + (q^2 + (c\delta)^2 - 2cq\delta)\alpha\beta\delta\\ &= 2cq\delta(q - c\delta) + (q - c\delta)^2\alpha\beta\delta \qquad \text{(B.47)}\end{aligned}$$

Substituting equation (B.47) into equation (B.46) and grouping common terms, we obtain:

$$\begin{aligned}
\mathrm{nr}(u_{22}) &= \lambda_j \cdot \Bigg( -(1+c-\alpha\delta\lambda_j)(q-c\delta)^2 - \lambda_j \cdot \big(2cq\delta(q-c\delta) + (q-c\delta)^2\alpha\beta\delta\big)\Bigg) \\
&= \lambda_j \cdot \Bigg( -(1+c-c\delta\lambda_j)(q-c\delta)^2 - \lambda_j \cdot \big(2cq\delta(q-c\delta)\big)\Bigg) \\
&= -\lambda_j \cdot \Bigg( (1+c-c\delta\lambda_j)(q-c\delta)^2 + 2cq\delta\lambda_j(q-c\delta)\Bigg) \qquad \text{(B.48)}
\end{aligned}$$

With this, we can write out the exact expression for $u_{22}$:

$$u_{22} = \frac{\big(1+c-c\delta\lambda_j\big)(q-c\delta) + 2cq\delta\lambda_j}{2\cdot(1-c^2+c\lambda_j\cdot(q+c\delta))} \qquad \text{(B.49)}$$

**Numerator of $u_{12}$**: We begin by rewriting the numerator of $u_{12}$ (from equation (B.39)):

$$\mathrm{nr}(u_{12}) = \lambda_j \cdot \Bigg( 2q\delta\big((1+c)(q\lambda_j - c) + \delta\lambda_j c^2\big) - (q^2+c^2\delta^2)\big(\lambda_j q - (1+c)(1-\delta\lambda_j)\big)\Bigg) \qquad \text{(B.50)}$$

We split the simplification into two parts: one depending on $(1+c)$ and the other part representing terms that don't contain $(1+c)$. In particular, we consider the terms that do not carry a coefficient of $(1+c)$:

$$\begin{aligned}
&2q\delta^2\lambda_j c^2 - (q^2+c^2\delta^2)\cdot(q\lambda_j) \\
&= q\lambda_j \cdot (2\delta^2c^2 - q^2 - \delta^2c^2) \\
&= -q\lambda_j \cdot (q^2 - (c\delta)^2) \qquad \text{(B.51)}
\end{aligned}$$

Next, we consider the other term containing the $(1+c)$ part:

$$\begin{aligned}
&(1+c)\cdot\Bigg(2q\delta\cdot(q\lambda_j - c) \ + \ (q^2+(c\delta)^2)\cdot(1-\delta\lambda_j)\Bigg) \\
&= (1+c)\cdot\Bigg(2q^2\delta\lambda_j - 2q\delta c + q^2 + (c\delta)^2 - q^2\delta\lambda_j - c^2\delta^3\lambda_j\Bigg) \\
&= (1+c)\cdot\Bigg( \ (q-c\delta)^2 + \delta\lambda_j \ (q^2-(c\delta)^2) \ \Bigg) \qquad \text{(B.52)}
\end{aligned}$$

Substituting equations (B.51), (B.52) into equation (B.50), we get:

$$\mathrm{nr}(u_{12}) = \lambda_j \cdot \big((1+c)\delta\lambda_j(q^2-(c\delta)^2) + (1+c)(q-c\delta)^2 - q\lambda_j(q^2-(c\delta)^2)\big)$$

$$
\begin{aligned}
&= \lambda_j \cdot \big((1+c)(q-c\delta)^2 + \lambda_j\big((1+c)\delta - q\big)\cdot(q^2-(c\delta)^2)\big)\\
&= \lambda_j \cdot \big((1+c)(q-c\delta)^2 + \lambda_j\big(\delta - (q-c\delta)\big)\cdot(q^2-(c\delta)^2)\big)\\
&= \lambda_j \cdot \big((1+c)(q-c\delta)^2 + \delta\lambda_j\cdot(q^2-(c\delta)^2) - \lambda_j(q+c\delta)(q-c\delta)^2\big)\\
&= \lambda_j \cdot \big((1+c-\lambda_j\cdot(q+c\delta))\cdot(q-c\delta)^2 + \delta\lambda_j\cdot(q^2-(c\delta)^2)\big) \qquad \text{(B.53)}
\end{aligned}
$$

With which, we can now write out the expression for $u_{12}$:

$$
u_{12} = \frac{\big(1+c-\lambda_j(q+c\delta)\big)(q-c\delta) + \delta\lambda_j(q+c\delta)}{2\cdot(1-c^2+c\lambda_j\cdot(q+c\delta))} \qquad \text{(B.54)}
$$

**Obtaining $u_{11}$:** We revisit equation (B.35) and substitute $u_{22}$ from equation (B.49):

$$
\begin{aligned}
u_{11} &= u_{22}(1-2\delta\lambda_j) + \delta^2\lambda_j\\
&= \frac{\big(1+c-c\delta\lambda_j\big)(q-c\delta) + 2cq\delta\lambda_j}{2\cdot(1-c^2+c\lambda_j\cdot(q+c\delta))}\cdot(1-2\delta\lambda_j) + \delta^2\lambda_j
\end{aligned}
$$

From which, we consider the numerator of $u_{11}$ and begin simplifying it:

$$
\begin{aligned}
\mathrm{nr}(u_{11}) &= (1+c-c\delta\lambda_j)(q-c\delta)(1-2\delta\lambda_j) + 2cq\delta\lambda_j(1-2\delta\lambda_j) + 2\delta^2\lambda_j(1-c^2+c\lambda_j(q+c\delta))\\
&= (1+c-c\delta\lambda_j)(q-c\delta)(1-2\delta\lambda_j) + 2\delta^2\lambda_j + 2c\delta\lambda_j(q-c\delta)(1-\delta\lambda_j)\\
&= (1+c+c\delta\lambda_j)(q-c\delta)(1-\delta\lambda_j) + 2\delta^2\lambda_j - \delta\lambda_j(1+c-c\delta\lambda_j)(q-c\delta)\\
&= (1+c+c\delta\lambda_j)(q-c\delta) - 2\delta\lambda_j(q-c\delta)(1+c) + 2\delta^2\lambda_j\\
&= (1+c-c\delta\lambda_j)(q-c\delta) - 2\delta\lambda_j(q-c\delta) + 2\delta^2\lambda_j \qquad \text{(B.55)}
\end{aligned}
$$

This implies,

$$
u_{11} = \frac{(1+c-c\delta\lambda_j)(q-c\delta) - 2\delta\lambda_j(q-c\delta) + 2\delta^2\lambda_j}{2\cdot(1-c^2+c\lambda_j\cdot(q+c\delta))} \qquad \text{(B.56)}
$$

**Obtaining a bound on $\mathbf{U}_{22}$**

For obtaining a PSD upper bound on $\mathbf{U}_{22}$, we will write out a sharp bound of $u_{22}$ in each eigen space:

$$
u_{22} = \frac{\big(1+c-c\lambda_j\delta\big)(q-c\delta) + 2cq\delta\lambda_j}{2\cdot(1-c^2+c\lambda_j\cdot(q+c\delta))}
$$

$$\begin{aligned}
&= \frac{\big(1 - c^2 + c\lambda_j(q + c\delta) + q\lambda_j + (1 + c)(c - \lambda_j(q + c\delta))\big)(q - c\delta) + 2cq\delta\lambda_j}{2 \cdot (1 - c^2 + c\lambda_j \cdot (q + c\delta))} \\
&= \frac{q - c\delta}{2} + \frac{q\lambda_j(q - c\delta)}{2 \cdot (1 - c^2 + c\lambda_j \cdot (q + c\delta))} + \frac{(1 + c)(c - \lambda_j(q + c\delta))(q - c\delta) + 2cq\delta\lambda_j}{2 \cdot (1 - c^2 + c\lambda_j \cdot (q + c\delta))} \\
&\le \frac{q - c\delta}{2} + \frac{q\lambda_j(q - c\delta)}{2 \cdot (c\lambda_j \cdot (q + c\delta))} + \frac{(1 + c)(c - \lambda_j(q + c\delta))(q - c\delta) + 2cq\delta\lambda_j}{2 \cdot (1 - c^2 + c\lambda_j \cdot (q + c\delta))} \\
&\le \frac{q - c\delta}{2} \cdot \frac{1 + c}{c} + \frac{(1 + c)(c - \lambda_j(q + c\delta))(q - c\delta) + 2cq\delta\lambda_j}{2 \cdot (1 - c^2 + c\lambda_j \cdot (q + c\delta))}
\end{aligned}$$

Let us consider bounding the numerator of the 2nd term:

$$\begin{aligned}
&(1 + c)(c - \lambda_j(q + c\delta))(q - c\delta) + 2cq\delta\lambda_j \\
&= c(1 + c)(q - c\delta) - (1 + c)\lambda_j(q + c\delta)(q - c\delta) + 2cq\delta\lambda_j \\
&= c(1 + c)(q - c\delta) - (1 + c)\lambda_j(q - c\delta)^2 - 2c\delta\lambda_j(1 + c)(q - c\delta) + 2cq\delta\lambda_j \\
&= c(1 + c)(q - c\delta) - (1 + c)\lambda_j(q - c\delta)^2 - 2c\delta\lambda_j(1 + c)(q - c\delta) + 2c(q - c\delta)\delta\lambda_j + 2c^2\delta^2\lambda_j \\
&= c(1 + c)(q - c\delta) + 2c^2\delta^2\lambda_j - (1 + c)\lambda_j(q - c\delta)^2 - 2c^2\delta\lambda_j(q - c\delta) \\
&\le c(1 + c)(q - c\delta) + 2c^2\delta^2\lambda_j
\end{aligned}$$

Implying,

$$\begin{aligned}
u_{22} &\le \frac{q - c\delta}{2} \cdot \frac{1 + c}{c} + \frac{c(1 + c)(q - c\delta) + 2c^2\delta^2\lambda_j}{2 \cdot (1 - c^2 + c\lambda_j \cdot (q + c\delta))} \\
&\le \frac{q - c\delta}{2} \cdot \frac{1 + c}{c} + \frac{c(1 + c)(q - c\delta)}{2 \cdot (1 - c^2 + c\lambda_j \cdot (q + c\delta))} + \frac{c^2\delta^2\lambda_j}{(1 - c^2 + c\lambda_j \cdot (q + c\delta))}
\end{aligned}$$

We will first upper bound the third term:

$$\begin{aligned}
\frac{c^2\delta^2\lambda_j}{(1 - c^2 + c\lambda_j \cdot (q + c\delta))} &\le \frac{c\delta^2}{(q + c\delta)} \\
&= \frac{c\delta^2}{(q - c\delta + 2c\delta)} \\
&\le \frac{c\delta^2}{2c\delta} = \frac{\delta}{2}
\end{aligned}$$

This implies,

$$u_{22} \le \frac{q - c\delta}{2} \cdot \frac{1 + c}{c} + \frac{\delta}{2} + \frac{c(1 + c)(q - c\delta)}{2 \cdot (1 - c^2 + c\lambda_j \cdot (q + c\delta))}$$

$$
\begin{aligned}
&= \frac{q-c\delta}{2}\cdot\frac{1+c}{c}+\frac{\delta}{2}+\frac{c^2(q-c\delta)}{1-c^2+c\lambda_j\cdot(q+c\delta)}+\frac{c(1-c)(q-c\delta)}{2\cdot(1-c^2+c\lambda_j\cdot(q+c\delta))}\\
&\le \frac{q-c\delta}{2}\cdot\frac{1+c}{c}+\frac{\delta}{2}+\frac{c^2(q-c\delta)}{1-c^2+c\lambda_j\cdot(q+c\delta)}+\frac{c(1-c)(q-c\delta)}{2\cdot(1-c^2)}\\
&= \frac{q-c\delta}{2}\cdot\frac{1+c}{c}+\frac{\delta}{2}+\frac{c^2(q-c\delta)}{1-c^2+c\lambda_j\cdot(q+c\delta)}+\frac{c(q-c\delta)}{2\cdot(1+c)}\\
&= \frac{q-c\delta}{2}\cdot\left(\frac{1+c}{c}+\frac{c}{1+c}\right)+\frac{\delta}{2}+\frac{c^2(q-c\delta)}{1-c^2+c\lambda_j\cdot(q+c\delta)}\\
&\le \frac{q-c\delta}{2}\cdot\frac{3}{c}+\frac{\delta}{2}+\frac{c^2(q-c\delta)}{1-c^2+c\lambda_j\cdot(q+c\delta)}\\
&\le \frac{q-c\delta}{2}\cdot\frac{3}{c}+\frac{\delta}{2}+\frac{c(q-c\delta)}{\lambda_j\cdot(q+c\delta)}\\
&= \frac{q-c\delta}{2}\cdot\frac{3}{c}+\frac{\delta}{2}+\frac{c(q-c\delta)}{\lambda_j\cdot(q-c\delta+2c\delta)}\\
&\le \frac{q-c\delta}{2}\cdot\frac{3}{c}+\frac{\delta}{2}+\frac{q-c\delta}{2\lambda_j\delta}\\
&\le \frac{4}{c}\cdot\frac{q-c\delta}{2\delta\lambda_j}+\frac{\delta}{2}
\end{aligned}
$$

Let us consider bounding $\frac{q-c\delta}{2\delta\lambda_j}$ :

$$
\frac{q-c\delta}{2\delta\lambda_j}=\frac{\alpha\beta\delta+\gamma(1-\alpha)}{2\delta\lambda_j}
$$

Substituting the values for $\alpha,\beta,\gamma,\delta$ applying $\frac{1}{1+\gamma\mu}\le 1$, $c_3=\frac{c_2\sqrt{2c_1-c_1^2}}{c_1}$ and, $c_2^2=\frac{c_4}{2-c_1}$ with $0<c_4<1/6$ we get:

$$
\begin{aligned}
\frac{q-c\delta}{2\delta\lambda_j} &\le \left(\frac{c_3c_2\sqrt{2c_1-c_1^2}}{2}\sqrt{\frac{\widetilde{\kappa}}{\kappa}}+\frac{c_2^2(2c_1-c_1^2)}{2c_1}\right)\cdot\frac{1}{\lambda_j\widetilde{\kappa}}\\
&\le \left(\frac{c_3c_2\sqrt{2c_1-c_1^2}}{2}+\frac{c_2^2(2c_1-c_1^2)}{2c_1}\right)\cdot\frac{1}{\lambda_j\widetilde{\kappa}}\\
&= c_2^2(2-c_1)\cdot\frac{1}{\widetilde{\kappa}\lambda_j}=c_4\cdot\frac{1}{\lambda_j\widetilde{\kappa}}
\end{aligned}
$$

Which implies the bound on $u_{22}$:

$$
u_{22}\le\frac{4}{c}\cdot\frac{c_4}{\lambda_j\widetilde{\kappa}}+\frac{\delta}{2}
$$

Now, consider the following bound on $1/c$:

$$\begin{aligned}\frac{1}{c} &= \frac{1}{\alpha(1-\beta)}\\ &= 1 + \frac{(1+c_3)c_2\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}} - c_2c_3\sqrt{2c_1-c_1^2}}\\ &\le 1 + \frac{(1+c_3)c_2\sqrt{2c_1-c_1^2}}{1 - c_2c_3\sqrt{2c_1-c_1^2}}\\ &= 1 + \frac{\sqrt{c_1c_4}+c_4}{1-c_4}\\ &= \frac{1+\sqrt{c_1c_4}}{1-c_4} \end{aligned} \tag{B.57}$$

Substituting values of $c_1$, $c_4$ we have: $1/c \le 1.5$. This implies the following bound on $u_{22}$:

$$u_{22} \le 6 \cdot \frac{c_4}{\lambda_j\widetilde{\kappa}} + \frac{\delta}{2} \tag{B.58}$$

Alternatively, this implies that $\mathbf{U}_{22}$ can be upper bounded in a psd sense as:

$$\mathbf{U}_{22} \preceq \frac{6c_4}{\widetilde{\kappa}} \cdot \mathbf{H}^{-1} + \frac{\delta}{2} \cdot \mathbf{I}$$

*Understanding fourth moment effects*

We wish to obtain a bound on:

$$\begin{aligned}\mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right]\mathbf{U} &= \mathbb{E}\left[\hat{\mathbf{V}}_2\mathbf{U}\hat{\mathbf{V}}_2^\top\right]\\ &= \begin{bmatrix}\delta^2 & \delta\cdot q\\ \delta\cdot q & q^2\end{bmatrix} \otimes \mathcal{M}\mathbf{U}_{22}\end{aligned}$$

We need to understand $\mathcal{M}\mathbf{U}_{22}$.

$$\begin{aligned}\mathcal{M}\mathbf{U}_{22} &\preceq \frac{6c_4}{\widetilde{\kappa}} \cdot \mathcal{M}\mathbf{H}^{-1} + \frac{\delta}{2}\cdot\mathcal{M}\mathbf{I}\\ &\preceq (6c_4 + \frac{\delta R^2}{2}) \cdot \mathbf{H}\\ &= s \cdot \mathbf{H}\end{aligned} \tag{B.59}$$

where, $s \overset{\text{def}}{=} (6c_4 + \frac{\delta R^2}{2}) = 23/30 \le \frac{4}{5}$. This implies (along with the fact that for any PSD matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, if $\mathbf{A} \preceq \mathbf{B}$, then, $\mathbf{A} \otimes \mathbf{C} \preceq \mathbf{B} \otimes \mathbf{C}$)),

$$\mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right] \mathbf{U} \preceq s \cdot \begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{H} \preceq \frac{4}{5} \cdot \begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{H}. \tag{B.60}$$

This will lead us to obtaining a PSD upper bound on $\mathbf{\Phi}_\infty$, i.e., the proof of Lemma 14

*Proof of lemma 14.* We begin by recounting the expression for the steady state covariance operator $\mathbf{\Phi}_\infty$ and applying results derived from previous subsections:

$$\begin{aligned}
\mathbf{\Phi}_\infty &= (\mathcal{I} - \mathcal{B})^{-1}\widehat{\mathbf{\Sigma}} \\
&\preceq \sigma^2\mathbf{U} + \sigma^2(\mathcal{I} - \mathcal{B})^{-1} \cdot \mathbb{E}\left[\hat{\mathbf{V}}_2 \otimes \hat{\mathbf{V}}_2\right] \mathbf{U} \quad \text{(from equation (B.33))} \\
&\preceq \sigma^2\mathbf{U} + \frac{4}{5}\sigma^2(\mathcal{I} - \mathcal{B})^{-1}\left( \begin{bmatrix} \delta^2 & \delta \cdot q \\ \delta \cdot q & q^2 \end{bmatrix} \otimes \mathbf{H}\right) \quad \text{(from equation (B.60))} \\
&= \sigma^2\mathbf{U} + \frac{4}{5}(\mathcal{I} - \mathcal{B})^{-1}\widehat{\mathbf{\Sigma}} \\
&= \sigma^2\mathbf{U} + \frac{4}{5} \cdot \mathbf{\Phi}_\infty \\
\implies \mathbf{\Phi}_\infty &\preceq 5\sigma^2\mathbf{U}.
\end{aligned} \tag{B.61}$$

Now, given the upper bound provided by equation (B.61), we can now obtain a (mildly) looser upper PSD bound on $\mathbf{U}$ that is more interpretable, and this is by providing an upper bound on $\mathbf{U}_{11}$ and $\mathbf{U}_{22}$ by considering their magnitude along each eigen direction of $\mathbf{H}$. In particular, let us consider the max of $u_{11}$ and $u_{22}$ along the $j^{th}$ eigen direction (as implied by equations (B.56), (B.49)):

$$\begin{aligned}
\max(u_{11}, u_{22}) &= \frac{(1 + c - c\delta\lambda_j)(q - c\delta) + 2\delta^2\lambda_j}{2 \cdot (1 - c^2 + c\lambda_j \cdot (q + c\delta))} \\
&= \frac{(1 + c - c\delta\lambda_j)(q - c\delta) + 2\delta^2\lambda_j}{2 \cdot (1 - c^2 + c\lambda_j \cdot (q + c\delta))} \\
&= \frac{(1 + c - c\delta\lambda_j)(q - c\delta) + 2cq\lambda_j - 2cq\lambda_j + 2\delta^2\lambda_j}{2 \cdot (1 - c^2 + c\lambda_j \cdot (q + c\delta))} \\
&= u_{22} + \frac{-2cq\lambda_j + 2\delta^2\lambda_j}{2 \cdot (1 - c^2 + c\lambda_j \cdot (q + c\delta))}
\end{aligned}$$

$$\leq \frac{6c_4}{\widetilde{\kappa}\lambda_j} + \frac{\delta}{2} + \frac{\delta^2\lambda_j - cq\lambda_j}{(1 - c^2 + c\lambda_j \cdot (q + c\delta))} \quad \text{(using equation (B.58))}$$

This implies, we can now consider upper bounding the term in the equation above and this will yield us the result:

$$\begin{aligned}
\frac{\delta^2\lambda_j - cq\lambda_j}{(1 - c^2 + c\lambda_j \cdot (q + c\delta))} &\leq \frac{\delta^2\lambda_j - cq\lambda_j}{c\lambda_j \cdot (q + c\delta)} \\
&\leq \frac{\delta^2\lambda_j - cq\lambda_j}{2c^2\delta\lambda_j} \\
&= \frac{\delta^2\lambda_j - c(\alpha\delta + \gamma(1-\alpha))\lambda_j}{2c^2\delta\lambda_j} \\
&\leq \frac{\delta^2\lambda_j - c\alpha\delta\lambda_j}{2c^2\delta\lambda_j} = \frac{1 - c\alpha}{c^2} \cdot \frac{\delta}{2} \\
&= \Big(\frac{1-c}{c^2} + \frac{1-\alpha}{c}\Big) \cdot \frac{\delta}{2} \\
&= \Big(\frac{(1+c_3)(1-\alpha)}{c^2} + \frac{1-\alpha}{c}\Big) \cdot \frac{\delta}{2} \\
&= \frac{1-\alpha}{c}\Big(\frac{(1+c_3)}{c} + 1\Big) \cdot \frac{\delta}{2} \\
&\leq 3\frac{1-\alpha}{c} \cdot \frac{1}{c} \cdot \frac{\delta}{2} \\
&\leq 3\frac{1-\alpha}{c} \cdot \frac{1 + \sqrt{c_1c_4}}{1 - c_4} \cdot \frac{\delta}{2} \\
&= 3 \cdot \frac{c_1c_3}{\sqrt{\kappa\widetilde{\kappa}} - c_1c_3^2} \cdot \frac{1 + \sqrt{c_1c_4}}{1 - c_4} \cdot \frac{\delta}{2} \\
&\leq 3 \cdot \frac{c_1c_3}{1 - c_1c_3^2} \cdot \frac{1 + \sqrt{c_1c_4}}{1 - c_4} \cdot \frac{\delta}{2} \\
&\leq (2/3)\frac{\delta}{2}
\end{aligned}$$

Plugging this into the bound for $\max u_{11}, u_{22}$, we get:

$$\max(u_{11}, u_{22}) \leq \frac{6c_4}{\widetilde{\kappa}\lambda_j} + (5/3)\frac{\delta}{2} = (2/3)\frac{1}{\widetilde{\kappa}\lambda_j} + (5/3)\frac{\delta}{2}$$

This implies the bound written out in the lemma, that is,

$$\mathbf{U} \preceq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \Big(\frac{2}{3}\big(\frac{1}{\widetilde{\kappa}}\mathbf{H}^{-1}\big) + \frac{5}{6} \cdot (\delta\ \mathbf{I})\Big)$$

$\square$

**Lemma 41.**

$$\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \left( \mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top} \right) \cdot \mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right] \rangle \leq$$
$$\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \left( \mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top} \right) \cdot \mathbb{E}\left[\boldsymbol{\theta}_\infty \otimes \boldsymbol{\theta}_\infty\right] \rangle \leq 5\sigma^2 d.$$

*Where, $d$ is the dimension of the problem.*

Before proving Lemma 41, we note that the sequence of expected covariances of the centered parameters $\mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right]$ when initialized at the zero covariance (as in the case of variance analysis) only grows (in a psd sense) as a function of time and settles at the steady state covariance.

**Lemma 42.** *Let $\boldsymbol{\theta}_0 = 0$. Then, by running the stochastic process defined using the recursion as in equation* (B.29)*, the covariance of the resulting process is monotonically increasing until reaching the stationary covariance $\mathbb{E}\left[\boldsymbol{\theta}_\infty \otimes \boldsymbol{\theta}_\infty\right]$.*

*Proof.* As long as the process does not diverge (as defined by spectral norm bounds of the expected update $\mathcal{B} = \mathbb{E}\left[\hat{\mathbf{A}} \otimes \hat{\mathbf{A}}\right]$ being less than 1), the first-order Markovian process converges geometrically to its unique stationary distribution $\boldsymbol{\theta}_\infty \otimes \boldsymbol{\theta}_\infty$. In particular,

$$\begin{aligned} \mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right] &= \mathcal{B}\mathbb{E}\left[\boldsymbol{\theta}_{l-1} \otimes \boldsymbol{\theta}_{l-1}\right] + \widehat{\boldsymbol{\Sigma}} \\ &= (\sum_{k=0}^{l-1} \mathcal{B}^k)\widehat{\boldsymbol{\Sigma}} \end{aligned}$$

Thus implying the fact that

$$\mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right] = \mathbb{E}\left[\boldsymbol{\theta}_{l-1} \otimes \boldsymbol{\theta}_{l-1}\right] + \mathcal{B}^{l-1}\widehat{\boldsymbol{\Sigma}}$$

Owing to the PSD'ness of the operators in the equation above, the lemma concludes with the claim that $\mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right] \succeq \mathbb{E}\left[\boldsymbol{\theta}_{l-1} \otimes \boldsymbol{\theta}_{l-1}\right]$ □

Given these lemmas, we are now in a position to prove Lemma 41.

*Proof of Lemma 41.*

$$
\begin{aligned}
&\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \Big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\Big) \cdot \mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right]\rangle \\
&= \langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \Big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\Big)(\mathcal{I} - \mathcal{A}_{\mathcal{L}}\mathcal{A}_{\mathcal{R}}^{\top})^{-1}(\mathcal{I} - \mathcal{A}_{\mathcal{L}}\mathcal{A}_{\mathcal{R}}^{\top}) \cdot \mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right]\rangle \\
&= \langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \Big((\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}(\mathcal{I} - \mathcal{A}_{\mathcal{R}}^{\top})^{-1}\Big)(\mathcal{I} - \mathcal{A}_{\mathcal{L}}\mathcal{A}_{\mathcal{R}}^{\top}) \cdot \mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right]\rangle \quad \text{(using Lemma 35)} \\
&= \langle \Big((\mathcal{I} - \mathcal{A}_{\mathcal{L}}^{\top})^{-1}(\mathcal{I} - \mathcal{A}_{\mathcal{R}})^{-1}\Big) \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, (\mathcal{I} - \mathcal{A}_{\mathcal{L}}\mathcal{A}_{\mathcal{R}}^{\top}) \cdot \mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right]\rangle \\
&= \langle (\mathbf{I} - \mathbf{A}^{\top})^{-1} \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix} (\mathbf{I} - \mathbf{A})^{-1}, (\mathcal{I} - \mathcal{A}_{\mathcal{L}}\mathcal{A}_{\mathcal{R}}^{\top}) \cdot \mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right]\rangle \\
&= \langle (\mathbf{I} - \mathbf{A}^{\top})^{-1} \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix} (\mathbf{I} - \mathbf{A})^{-1}, (\mathcal{I} - \mathcal{D}) \cdot \mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right]\rangle \\
&= \frac{1}{(q - c\delta)^2}\langle \Big( \otimes_2 \begin{bmatrix} -(c\mathbf{I} - q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I} - \delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix} \Big), (\mathcal{I} - \mathcal{D}) \cdot \mathbb{E}\left[\boldsymbol{\theta}_l \otimes \boldsymbol{\theta}_l\right]\rangle \quad \text{(using Lemma 32)} \\
&= \frac{1}{(q - c\delta)^2}\langle \Big( \otimes_2 \begin{bmatrix} -(c\mathbf{I} - q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I} - \delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix} \Big), (\mathcal{I} - \mathcal{D})(\mathcal{I} - \mathcal{B})^{-1}(\mathcal{I} - \mathcal{B}^{l})\widehat{\boldsymbol{\Sigma}}\rangle \\
&= \frac{1}{(q - c\delta)^2}\langle \Big( \otimes_2 \begin{bmatrix} -(c\mathbf{I} - q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I} - \delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix} \Big), (\mathcal{I} - \mathcal{B} + \mathcal{R})(\mathcal{I} - \mathcal{B})^{-1}(\mathcal{I} - \mathcal{B}^{l})\widehat{\boldsymbol{\Sigma}}\rangle \\
&= \frac{1}{(q - c\delta)^2}\langle \Big( \otimes_2 \begin{bmatrix} -(c\mathbf{I} - q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I} - \delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix} \Big), \widehat{\boldsymbol{\Sigma}} - \mathcal{B}^{l}\widehat{\boldsymbol{\Sigma}} + \mathcal{R}(\mathcal{I} - \mathcal{B})^{-1}\widehat{\boldsymbol{\Sigma}} - \mathcal{R}(\mathcal{I} - \mathcal{B})^{-1}\mathcal{B}^{l}\widehat{\boldsymbol{\Sigma}}\rangle \\
&\leq \frac{1}{(q - c\delta)^2}\langle \Big( \otimes_2 \begin{bmatrix} -(c\mathbf{I} - q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I} - \delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix} \Big), \widehat{\boldsymbol{\Sigma}} + \sigma^2\mathcal{R} \cdot (5\mathbf{U})\rangle \qquad \text{(B.62)}
\end{aligned}
$$

So, we need to understand $\mathcal{R}\mathbf{U}$:

$$
\mathcal{R}\mathbf{U} = \mathbb{E}\Big( \begin{bmatrix} 0 & \delta \cdot (\mathbf{H} - \mathbf{a}\mathbf{a}^{\top}) \\ 0 & q \cdot (\mathbf{H} - \mathbf{a}\mathbf{a}^{\top}) \end{bmatrix} \mathbf{U} \begin{bmatrix} 0 & 0 \\ \delta \cdot (\mathbf{H} - \mathbf{a}\mathbf{a}^{\top}) & q \cdot (\mathbf{H} - \mathbf{a}\mathbf{a}^{\top}) \end{bmatrix} \Big)
$$

$$
\begin{aligned}
&= \begin{bmatrix} \delta^2 & \delta\cdot q \\ \delta\cdot q & q^2 \end{bmatrix} \otimes \mathbb{E}\left[(\mathbf{H}-\mathbf{a}\mathbf{a}^\top)\mathbf{U}_{22}(\mathbf{H}-\mathbf{a}\mathbf{a}^\top)\right] \\
&= \begin{bmatrix} \delta^2 & \delta\cdot q \\ \delta\cdot q & q^2 \end{bmatrix} \otimes \big(\mathcal{M}-\mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}}\big)\mathbf{U}_{22} \\
&\preceq \begin{bmatrix} \delta^2 & \delta\cdot q \\ \delta\cdot q & q^2 \end{bmatrix} \otimes \mathcal{M}\mathbf{U}_{22} \\
&\preceq \frac{4}{5}\cdot \begin{bmatrix} \delta^2 & \delta\cdot q \\ \delta\cdot q & q^2 \end{bmatrix} \otimes \mathbf{H} \quad \text{(from equation (B.59)).}
\end{aligned}
$$

Then,

$$
\begin{aligned}
&\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \Big(\mathcal{I}+(\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}}+(\mathcal{I}-\mathcal{A}_{\mathcal{R}}^\top)^{-1}\mathcal{A}_{\mathcal{R}}^\top\Big)\cdot\boldsymbol{\theta}_l\otimes\boldsymbol{\theta}_l\rangle \\
&\le \frac{1}{(q-c\delta)^2}\langle\Bigg(\otimes_2 \begin{bmatrix} -(c\mathbf{I}-q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Bigg), \widehat{\boldsymbol{\Sigma}}+\sigma^2\mathcal{R}\cdot(5\mathbf{U})\rangle \qquad \text{(from equation (B.62))} \\
&\le \frac{5\sigma^2}{(q-c\delta)^2}\cdot\langle\Bigg(\otimes_2 \begin{bmatrix} -(c\mathbf{I}-q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Bigg), \begin{bmatrix} \delta^2 & \delta\cdot q \\ \delta\cdot q & q^2 \end{bmatrix}\otimes\mathbf{H}\rangle \\
&= \frac{5}{(q-c\delta)^2}\cdot d\,\sigma^2\cdot(q-c\delta)^2 \\
&= 5\sigma^2 d. 
\end{aligned} \tag{B.63}
$$

□

**Lemma 43.**

$$
\Bigg|\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \Big((\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-2}\mathcal{A}_{\mathcal{L}}+(\mathcal{I}-\mathcal{A}_{\mathcal{R}}^\top)^{-2}\mathcal{A}_{\mathcal{R}}^\top\Big)\boldsymbol{\Phi}_\infty\rangle\Bigg| \le C\cdot\sigma^2 d\sqrt{\kappa\widetilde{\kappa}}
$$

*Where, $C$ is a universal constant.*

*Proof.* We begin by noting the following while considering the left side of the above expression:

$$\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \Big( (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^\top)^{-2} \mathcal{A}_{\mathcal{R}}^\top + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-2} \mathcal{A}_{\mathcal{L}} \Big) \mathbf{\Phi}_\infty \rangle$$
$$= \langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix} \mathbf{A}(\mathbf{I} - \mathbf{A})^{-2} + (\mathbf{I} - \mathbf{A}^\top)^{-2} \mathbf{A}^\top \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{\Phi}_\infty \rangle$$

The inner product above is a sum of two terms, so let us consider the first of the terms:

$$\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix} \mathbf{A}(\mathbf{I} - \mathbf{A})^{-2}, \mathbf{\Phi}_\infty \rangle$$
$$= \mathrm{Tr}\Big( (\mathbf{I} - \mathbf{A}^\top)^{-2} \mathbf{A}^\top \begin{bmatrix} \mathbf{H}^{1/2} \\ 0 \end{bmatrix} \begin{bmatrix} \mathbf{H}^{1/2} & 0 \end{bmatrix} \mathbf{\Phi}_\infty \Big)$$
$$= \mathrm{Tr}\Big( \Big( \begin{bmatrix} \mathbf{H}^{1/2} \\ 0 \end{bmatrix}^\top \mathbf{\Phi}_\infty (\mathbf{I} - \mathbf{A}^\top)^{-2} \mathbf{A}^\top \begin{bmatrix} \mathbf{H}^{1/2} \\ 0 \end{bmatrix} \Big) \Big)$$
$$= \sum_{j=1}^{d} \mathrm{Tr}\Big( \Big( \begin{bmatrix} \lambda_j^{1/2} \\ 0 \end{bmatrix}^\top (\mathbf{\Phi}_\infty)_j (\mathbf{I} - \mathbf{A}_j^\top)^{-2} \mathbf{A}_j^\top \begin{bmatrix} \lambda_j^{1/2} \\ 0 \end{bmatrix} \Big) \Big)$$
$$= \sum_{j=1}^{d} \mathrm{Tr}\Big( \Big( \begin{bmatrix} \lambda_j^{1/2} \\ 0 \end{bmatrix}^\top (\mathbf{\Phi}_\infty^{1/2})_j \Big) \cdot \Big( (\mathbf{\Phi}_\infty^{1/2})_j^\top (\mathbf{I} - \mathbf{A}_j^\top)^{-2} \mathbf{A}_j^\top \begin{bmatrix} \lambda_j^{1/2} \\ 0 \end{bmatrix} \Big) \Big),$$

where $(\mathbf{\Phi}_\infty)_j$ is the $2 \times 2$ block of $\mathbf{\Phi}_\infty$ corresponding to the $j^{\text{th}}$ eigensubspace of $\mathbf{H}$, $(\mathbf{\Phi}_\infty^{1/2})_j$ denotes the $2 \times 2d$ submatrix (i.e., 2 rows) of $\mathbf{\Phi}_\infty^{1/2}$ corresponding to the $j^{\text{th}}$ eigensubspace and $\mathbf{A}_j$ denotes the $j^{\text{th}}$ diagonal block of $\mathbf{A}$. Note that $(\mathbf{\Phi}_\infty^{1/2})_j (\mathbf{\Phi}_\infty^{1/2})_j^\top = (\mathbf{\Phi}_\infty)_j$. It is very easy to observe that the second term in the dot product can be written in a similar manner, i.e.:

$$\langle (\mathbf{I} - \mathbf{A}^\top)^{-2} \mathbf{A}^\top \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{\Phi}_\infty \rangle$$
$$= \sum_{j=1}^{d} \mathrm{Tr}\Big( \Big( (\mathbf{\Phi}_\infty^{1/2})_j^\top \begin{bmatrix} \lambda_j^{1/2} \\ 0 \end{bmatrix} \Big) \cdot \Big( \begin{bmatrix} \lambda_j^{1/2} \\ 0 \end{bmatrix}^\top \mathbf{A}_j (\mathbf{I} - \mathbf{A}_j)^{-2} (\mathbf{\Phi}_\infty^{1/2})_j \Big) \Big)$$

So, essentially, the expression in the left side of the lemma can be upper bounded by using Cauchy-Shwartz inequality:

$$\begin{aligned}
&\operatorname{Tr}\Bigg(\Bigg(\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}^\top (\mathbf{\Phi}_\infty^{1/2})_j\Bigg)\cdot\Bigg((\mathbf{\Phi}_\infty^{1/2})_j^\top(\mathbf{I}-\mathbf{A}_j^\top)^{-2}\mathbf{A}_j^\top\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}\Bigg)\Bigg)\\
&+\operatorname{Tr}\Bigg(\Bigg((\mathbf{\Phi}_\infty^{1/2})_j^\top\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}\Bigg)\cdot\Bigg(\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}^\top\mathbf{A}_j(\mathbf{I}-\mathbf{A}_j)^{-2}(\mathbf{\Phi}_\infty^{1/2})_j\Bigg)\Bigg)\\
&\le 2\left\|\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}\right\|_{(\mathbf{\Phi}_\infty)_j}\cdot\left\|(\mathbf{I}-\mathbf{A}_j^\top)^{-2}\mathbf{A}_j^\top\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}\right\|_{(\mathbf{\Phi}_\infty)_j}
\end{aligned}\tag{B.64}$$

The advantage with the above expression is that we can now begin to employ psd upper bounds on the covariance of the steady state distribution $\mathbf{\Phi}_\infty$ and provide upper bounds on the expression on the right hand side. In particular, we employ the following bound provided by the taylor expansion that gives us an upper bound on $\mathbf{\Phi}_\infty$:

$$\mathbf{\Phi}_\infty \overset{\text{def}}{=} \begin{bmatrix}\hat{\mathbf{U}}_{11} & \hat{\mathbf{U}}_{12}\\ \hat{\mathbf{U}}_{12}^\top & \hat{\mathbf{U}}_{22}\end{bmatrix} \preceq 5\sigma^2\mathbf{U} = 5\sigma^2\begin{bmatrix}\mathbf{U}_{11} & \mathbf{U}_{12}\\ \mathbf{U}_{12}^\top & \mathbf{U}_{22}\end{bmatrix}\qquad\text{(using equation (B.61))}$$

This implies in particular that $(\mathbf{\Phi}_\infty)_j \preceq 5\sigma^2\mathbf{U}_j$ for every $j\in[d]$ and hence, for any vector $\|\mathbf{a}\|_{(\mathbf{\Phi}_\infty)_j} \le \sqrt{5\sigma^2}\,\|\mathbf{a}\|_{\mathbf{U}_j}$. The important property of the matrix $\mathbf{U}$ that serves as a PSD upper bound is that it is diagonalizable using the basis of $\mathbf{H}$, thus allowing us to bound the computations in each of the eigen directions of $\mathbf{H}$.

$$\begin{aligned}
&\left\|(\mathbf{I}-\mathbf{A}_j^\top)^{-2}\mathbf{A}_j^\top\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}\right\|_{(\mathbf{\Phi}_\infty)_j}\\
&=\sqrt{\begin{bmatrix}\lambda_j^{1/2} & 0\end{bmatrix}\mathbf{A}_j(\mathbf{I}-\mathbf{A}_j)^{-2}(\mathbf{\Phi}_\infty)_j(\mathbf{I}-\mathbf{A}_j^\top)^{-2}\mathbf{A}_j^\top\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}}\\
&\le\sqrt{5\sigma^2\begin{bmatrix}\lambda_j^{1/2} & 0\end{bmatrix}\mathbf{A}_j(\mathbf{I}-\mathbf{A}_j)^{-2}\mathbf{U}_j(\mathbf{I}-\mathbf{A}_j^\top)^{-2}\mathbf{A}_j^\top\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}}
\end{aligned}$$

$$=\sqrt{5\sigma^2}\left\|(\mathbf{I}-\mathbf{A}_j^\top)^{-2}\mathbf{A}_j^\top\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}\right\|_{\mathbf{U}_j} \tag{B.65}$$

So, let us consider $\begin{bmatrix}\lambda_j^{1/2} & 0\end{bmatrix}\mathbf{A}_j(\mathbf{I}-\mathbf{A}_j)^{-2}$ and write out the following series of equations:

$$\begin{aligned}
\begin{bmatrix}\lambda_j^{1/2} & 0\end{bmatrix}\mathbf{A}_j &= \begin{bmatrix}0 & \sqrt{\lambda_j}(1-\delta\lambda_j)\end{bmatrix}\\
\mathbf{I}-\mathbf{A}_j &= \begin{bmatrix}1 & -(1-\delta\lambda_j)\\ c & -(c-q\lambda_j)\end{bmatrix}\\
\det(\mathbf{I}-\mathbf{A}_j) &= (q-c\delta)\lambda_j\\
(\mathbf{I}-\mathbf{A}_j)^{-1} &= \frac{1}{(q-c\delta)\lambda_j}\begin{bmatrix}-(c-q\lambda_j) & 1-\delta\lambda_j\\ -c & 1\end{bmatrix}\\
\implies \begin{bmatrix}\lambda_j^{1/2} & 0\end{bmatrix}\mathbf{A}_j(\mathbf{I}-\mathbf{A}_j)^{-1} &= \frac{\sqrt{\lambda_j}(1-\delta\lambda_j)}{(q-c\delta)\lambda_j}\begin{bmatrix}-c & 1\end{bmatrix}\\
\implies \begin{bmatrix}\lambda_j^{1/2} & 0\end{bmatrix}\mathbf{A}_j(\mathbf{I}-\mathbf{A}_j)^{-2} &= \frac{\sqrt{\lambda_j}(1-\delta\lambda_j)}{((q-c\delta)\lambda_j)^2}\begin{bmatrix}-c(1-c+q\lambda_j) & 1-c+c\delta\lambda_j\end{bmatrix}\\
&= \frac{\sqrt{\lambda_j}(1-\delta\lambda_j)}{((q-c\delta)\lambda_j)^2}\\
&\quad\cdot\left((1-c+c\delta\lambda_j)\begin{bmatrix}-c & 1\end{bmatrix} - c\lambda_j(q-c\delta)\begin{bmatrix}1 & 0\end{bmatrix}\right)
\end{aligned}$$

This implies,

$$\begin{aligned}
\left\|(\mathbf{I}-\mathbf{A}_j^\top)^{-2}\mathbf{A}_j^\top\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}\right\|_{\mathbf{U}_j} \le{}& \frac{\sqrt{\lambda_j}(1-\delta\lambda_j)}{((q-c\delta)\lambda_j)^2}\cdot(1-c+c\delta\lambda_j)\left\|\begin{bmatrix}-c\\1\end{bmatrix}\right\|_{\mathbf{U}_j}\\
&+\frac{c\sqrt{\lambda_j}(1-\delta\lambda_j)}{((q-c\delta)\lambda_j)}\left\|\begin{bmatrix}1\\0\end{bmatrix}\right\|_{\mathbf{U}_j}
\end{aligned} \tag{B.66}$$

Next, let us consider $\left\|\begin{bmatrix}-c\\1\end{bmatrix}\right\|_{\mathbf{U}_j}^2$:

$$\left\|\begin{bmatrix}-c\\1\end{bmatrix}\right\|_{\mathbf{U}_j}^2 = c^2u_{11}+u_{22}-2c\cdot u_{12}$$

Note that $u_{11}, u_{12}, u_{22}$ share the same denominator, so let us evaluate the numerator $\mathrm{nr}(c^2u_{11} - 2cu_{12} + u_{22})$. For this, we have, from equations (B.56), (B.54), (B.49) respectively: Furthermore,

$$\begin{aligned}
\mathrm{nr}(u_{11}) &= (1 + c - c\delta\lambda_j)(q - c\delta) - 2\delta\lambda_j(q - c\delta) + 2\delta^2\lambda_j \\
\mathrm{nr}(u_{12}) &= (1 + c - \lambda_j(q + c\delta))(q - c\delta) + \delta\lambda_j(q + c\delta) \\
\mathrm{nr}(u_{22}) &= (1 + c - c\delta\lambda_j)(q - c\delta) + 2cq\delta\lambda_j
\end{aligned}$$

Combining these, we have:

$$\begin{aligned}
&c^2\mathrm{nr}(u_{11}) - 2c \cdot \mathrm{nr}(u_{12}) + \mathrm{nr}(u_{22}) \\
=&\big((1 + c - c\delta\lambda_j)(1 - c)^2 + 2cq\lambda_j\big)(q - c\delta) - 2c^2\delta\lambda_j(q - c\delta) \\
=&\big((1 + c - c\delta\lambda_j)(1 - c)^2(q - c\delta)\big) + 2c\lambda_j(q - c\delta)^2
\end{aligned}$$

Implying,

$$\left\| \begin{bmatrix} -c \\ 1 \end{bmatrix} \right\|_{\mathbf{U}_j}^2 = \frac{(1 + c - c\delta\lambda_j)(1 - c)^2(q - c\delta) + 2c\lambda_j(q - c\delta)^2}{1 - c^2 + c\lambda_j(q + c\delta)}$$

In a very similar manner,

$$\begin{aligned}
\left\| \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\|_{\mathbf{U}_j}^2 &= u_{11} \\
&= \frac{(1 + c - c\delta\lambda_j)(q - c\delta) - 2\delta\lambda_j(q - c\delta) + 2\delta^2\lambda_j}{1 - c^2 + c\lambda_j(q + c\delta)}
\end{aligned}$$

This implies, plugging into equation (B.66)

$$\begin{aligned}
&\left\| (\mathbf{I} - \mathbf{A}_j^\top)^{-2}\mathbf{A}_j^\top \begin{bmatrix} \lambda_j^{1/2} \\ 0 \end{bmatrix} \right\|_{\mathbf{U}_j} \\
\le& \frac{\sqrt{\lambda_j}(1 - \delta\lambda_j)}{((q - c\delta)\lambda_j)^2} \cdot (1 - c + c\delta\lambda_j)\sqrt{\frac{(1 + c - c\delta\lambda_j)(1 - c)^2(q - c\delta) + 2c\lambda_j(q - c\delta)^2}{1 - c^2 + c\lambda_j(q + c\delta)}} \\
+& \frac{c\sqrt{\lambda_j}(1 - \delta\lambda_j)}{((q - c\delta)\lambda_j)}\sqrt{\frac{(1 + c - c\delta\lambda_j)(q - c\delta) - 2\delta\lambda_j(q - c\delta) + 2\delta^2\lambda_j}{1 - c^2 + c\lambda_j(q + c\delta)}} \qquad \text{(B.67)}
\end{aligned}$$

Finally, we need,

$$\left\|\begin{bmatrix}\mathbf{H}^{1/2}\\ 0\end{bmatrix}\right\|_{\mathbf{\Phi}_\infty} \leq \sqrt{5\sigma^2}\left\|\begin{bmatrix}\mathbf{H}^{1/2}\\ 0\end{bmatrix}\right\|_{\mathbf{U}}$$

Again, this can be analyzed in each of the eigen directions $(\lambda_j, \mathbf{u}_j)$ of $\mathbf{H}$ to yield:

$$\begin{aligned}\left\|\begin{bmatrix}\lambda_j^{1/2}\\ 0\end{bmatrix}\right\|_{\mathbf{U}_j} &= \sqrt{\lambda_j u_{11}}\\ &= \sqrt{\lambda_j \cdot \frac{(1+c-c\delta\lambda_j)(q-c\delta) - 2\delta\lambda_j(q-c\delta) + 2\delta^2\lambda_j}{1-c^2+c\lambda_j(q+c\delta)}} \qquad \text{(B.68)}\end{aligned}$$

Now, we require to bound the product of equation (B.67) and (B.68):

$$\left\|(\mathbf{I}-\mathbf{A}_j^\top)^{-2}\mathbf{A}_j^\top\begin{bmatrix}\lambda_j^{1/2}\\ 0\end{bmatrix}\right\|_{\mathbf{U}_j} \cdot \left\|\begin{bmatrix}\lambda_j^{1/2}\\ 0\end{bmatrix}\right\|_{\mathbf{U}_j} = T_1 + T_2 \qquad \text{(B.69)}$$

Where,

$$\begin{aligned}T_1 = &\frac{\lambda_j(1-\delta\lambda_j)}{((q-c\delta)\lambda_j)^2}\cdot(1-c+c\delta\lambda_j)\left(\sqrt{\frac{(1+c-c\delta\lambda_j)(1-c)^2(q-c\delta)+2c\lambda_j(q-c\delta)^2}{1-c^2+c\lambda_j(q+c\delta)}}\right)\\ &\cdot\left(\sqrt{\frac{(1+c-c\delta\lambda_j)(q-c\delta)-2\delta\lambda_j(q-c\delta)+2\delta^2\lambda_j}{1-c^2+c\lambda_j(q+c\delta)}}\right)\end{aligned}$$

And,

$$T_2 = \frac{c(1-\delta\lambda_j)}{q-c\delta}\cdot\left(\frac{(1+c-c\delta\lambda_j)(q-c\delta)-2\delta\lambda_j(q-c\delta)+2\delta^2\lambda_j}{1-c^2+c\lambda_j(q+c\delta)}\right)$$

We begin by considering $T_1$:

$$\begin{aligned}T_1 = &\frac{\lambda_j(1-\delta\lambda_j)}{((q-c\delta)\lambda_j)^2}\cdot(1-c+c\delta\lambda_j)\left(\sqrt{\frac{(1+c-c\delta\lambda_j)(1-c)^2(q-c\delta)+2c\lambda_j(q-c\delta)^2}{1-c^2+c\lambda_j(q+c\delta)}}\right)\\ &\cdot\left(\sqrt{\frac{(1+c-c\delta\lambda_j)(q-c\delta)-2\delta\lambda_j(q-c\delta)+2\delta^2\lambda_j}{1-c^2+c\lambda_j(q+c\delta)}}\right)\\ = &\left(\frac{\lambda_j(1-\delta\lambda_j)}{((q-c\delta)\lambda_j)^2}\right)\cdot\left(\frac{1-c+c\delta\lambda_j}{1-c^2+c\lambda_j(q+c\delta)}\right)\cdot\end{aligned}$$

$$
\begin{aligned}
&\Bigg(\sqrt{(1+c-c\delta\lambda_j)(q-c\delta)-2\delta\lambda_j(q-c\delta)+2\delta^2\lambda_j}\cdot \\
&\qquad\qquad\qquad \sqrt{(1+c-c\delta\lambda_j)(1-c)^2(q-c\delta)+2c\lambda_j(q-c\delta)^2}\Bigg) \\
&\leq \left(\frac{\lambda_j}{((q-c\delta)\lambda_j)^2}\right)\cdot\left(\frac{1-c+c\delta\lambda_j}{1-c^2+c\lambda_j(q+c\delta)}\right)\cdot \\
&\Bigg(\sqrt{(1+c-c\delta\lambda_j)(q-c\delta)+2\delta^2\lambda_j}\cdot\sqrt{(1+c-c\delta\lambda_j)(1-c)^2(q-c\delta)+2c\lambda_j(q-c\delta)^2}\Bigg)
\end{aligned}
\tag{B.70}
$$

We will consider the four terms within the square root and bound them separately:

$$
\begin{aligned}
T_1^{11} &= \frac{(1+c-c\delta\lambda_j)(1-c)}{(q-c\delta)\lambda_j} \\
&\leq \frac{2(1-c)}{\lambda_j\cdot(q-c\delta)} \leq \frac{2(1+c_3)}{\lambda_j\gamma} \\
&\leq \frac{2(1+c_3)}{c_2\sqrt{2c_1-c_1^2}}\sqrt{\kappa\widetilde{\kappa}}
\end{aligned}
$$

Next,

$$
\begin{aligned}
T_1^{21} &= \frac{\sqrt{2\delta^2\lambda_j}\sqrt{(1+c-c\delta\lambda_j)(1-c)^2(q-c\delta)}}{(q-c\delta)^2\lambda_j} \\
&\leq \frac{2\delta(1-c)}{\sqrt{(q-c\delta)^3\lambda_j}} = \frac{2\delta}{\sqrt{(q-c\delta)\lambda_j}}\frac{1-c}{q-c\delta} \\
&= \frac{2(1+c_3)\delta}{\gamma}\cdot\frac{1}{\sqrt{(q-c\delta)\lambda_j}} \\
&\leq \frac{2(1+c_3)\delta}{\gamma}\cdot\frac{1}{\sqrt{\gamma(1-\alpha)\mu}} \\
&\leq \frac{2\sqrt{2}(1+c_3)}{c_2^2(2-c_1)}\cdot\widetilde{\kappa}
\end{aligned}
$$

Next,

$$
\begin{aligned}
T_1^{12} &= \frac{\sqrt{(1+c-c\delta\lambda_j)(q-c\delta)^3\cdot 2c\lambda_j}}{(q-c\delta)^2\lambda_j} \\
&\leq \frac{2\sqrt{2}}{c_2\sqrt{2c_1-c_1^2}}\cdot\sqrt{\kappa\widetilde{\kappa}}
\end{aligned}
$$

Finally,

$$\begin{aligned}
T_1^{22} &= \frac{\sqrt{2\delta^2\lambda_j \cdot 2\ c\lambda_j(q-c\delta)^2}}{(q-c\delta)^2\lambda_j} \\
&\leq \frac{2\delta}{q-c\delta} \leq \frac{4}{c_2^2(2-c_1)} \cdot \widetilde{\kappa}
\end{aligned}$$

Implying,

$$\begin{aligned}
T_1 &\leq \left(\frac{1-c+c\delta\lambda_j}{1-c^2+c\lambda_j(q+c\delta)}\right) \cdot (T_1^{11}+T_1^{12}+T_1^{21}+T_1^{22}) \\
&\leq \left(\frac{1-c+c\delta\lambda_j}{1-c^2+c\lambda_j(q+c\delta)}\right) \cdot 2 \cdot (1+\sqrt{2}+c_3)\left(\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_2\sqrt{2c_1-c_1^2}} + \sqrt{2}\frac{\widetilde{\kappa}}{c_2^2(2-c_1)}\right) \\
&\leq \left(\frac{1}{1+c}+\frac{1}{2c}\right) \cdot 2 \cdot (1+\sqrt{2}+c_3)\left(\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_2\sqrt{2c_1-c_1^2}} + \sqrt{2}\frac{\widetilde{\kappa}}{c_2^2(2-c_1)}\right) \\
&= \left(\frac{1}{1+c}+\frac{1}{2c}\right) \cdot 2 \cdot (1+\sqrt{2}+c_3)\left(\frac{\sqrt{\kappa\widetilde{\kappa}}}{\sqrt{c_1c_4}} + \frac{\sqrt{2}\widetilde{\kappa}}{c_4}\right) \\
&\leq \frac{3}{c} \cdot (1+\sqrt{2}+c_3)\left(\frac{\sqrt{\kappa\widetilde{\kappa}}}{\sqrt{c_1c_4}} + \frac{\sqrt{2}\widetilde{\kappa}}{c_4}\right)
\end{aligned}$$

Recall the bound on 1/c from equation (B.57):

$$\frac{1}{c} \leq \frac{1+\sqrt{c_1c_4}}{1-c_4}$$

Implying,

$$\begin{aligned}
T_1 &\leq \frac{3}{c} \cdot (1+\sqrt{2}+c_3)\left(\frac{\sqrt{\kappa\widetilde{\kappa}}}{\sqrt{c_1c_4}} + \frac{\sqrt{2}\widetilde{\kappa}}{c_4}\right) \\
&\leq \frac{3}{c} \cdot (1+\sqrt{2}+c_3)\left(\frac{1}{\sqrt{c_1c_4}} + \frac{\sqrt{2}}{c_4}\right)\sqrt{\kappa\widetilde{\kappa}} \\
&\leq 3(1+\sqrt{2}+c_3)\left(\frac{1}{\sqrt{c_1c_4}} + \frac{\sqrt{2}}{c_4}\right) \cdot \frac{1+\sqrt{c_1c_4}}{1-c_4}\sqrt{\kappa\widetilde{\kappa}} \\
&\leq 3(1+\sqrt{2}+\sqrt{(c_4/c_1)})\left(\frac{1}{\sqrt{c_1c_4}} + \frac{\sqrt{2}}{c_4}\right) \cdot \frac{1+\sqrt{c_1c_4}}{1-c_4}\sqrt{\kappa\widetilde{\kappa}} \qquad \text{(B.71)}
\end{aligned}$$

Next, we consider $T_2$:

$$T_2 = \frac{c(1-\delta\lambda_j)}{q-c\delta} \cdot \left(\frac{(1+c-c\delta\lambda_j)(q-c\delta) - 2\delta\lambda_j(q-c\delta) + 2\delta^2\lambda_j}{1-c^2+c\lambda_j(q+c\delta)}\right)$$

$$\leq \left(\frac{(1+c-c\delta\lambda_j)(q-c\delta)-2\delta\lambda_j(q-c\delta)+2\delta^2\lambda_j}{(q-c\delta)\cdot(1-c^2+c\lambda_j(q+c\delta))}\right)$$
$$\leq \left(\frac{(1+c-c\delta\lambda_j)(q-c\delta)+2\delta^2\lambda_j}{(q-c\delta)\cdot(1-c^2+c\lambda_j(q+c\delta))}\right)$$

We split $T_2$ into two parts:

$$\begin{aligned}
T_2^1 &= \frac{(1+c-c\delta\lambda_j)}{(1-c^2+c\lambda_j(q+c\delta))}\\
&\leq \frac{1}{1-c} = \frac{1}{1-\alpha+\alpha\beta}\\
&= \frac{1}{(1+c_3)(1-\alpha)}\\
&\leq \frac{2\sqrt{\kappa\widetilde{\kappa}}}{(1+c_3)c_2\sqrt{2c_1-c_1^2}}\\
&\leq \frac{2\sqrt{\kappa\widetilde{\kappa}}}{(1+\sqrt{c_4/c_1})\sqrt{c_1c_4}}\\
&= \frac{2\sqrt{\kappa\widetilde{\kappa}}}{\sqrt{c_1c_4}+c_4}
\end{aligned}$$

Then,

$$\begin{aligned}
T_2^2 &= \frac{2\delta^2\lambda_j}{(q-c\delta)(1-c^2+c\lambda_j(q+c\delta))}\\
&\leq \frac{\delta^2\lambda_j}{\gamma(1-\alpha)c^2\lambda_j\delta} = \frac{\delta}{c^2\gamma(1-\alpha)}\\
&= \frac{2\widetilde{\kappa}}{c_4}\cdot\frac{1}{c^2}
\end{aligned}$$

Implying,

$$\begin{aligned}
T_2 &\leq 2\cdot\left(\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_4+\sqrt{c_1c_4}}+\frac{\widetilde{\kappa}}{c^2c_4}\right)\\
&\leq 2\cdot\left(\frac{1}{\sqrt{c_1c_4}+c_4}+(\frac{1+\sqrt{c_1c_4}}{1-c_4})^2\cdot\frac{1}{c_4}\right)\sqrt{\kappa\widetilde{\kappa}}\\
&\leq \frac{2}{c_4}\cdot\left(1+(\frac{1+\sqrt{c_1c_4}}{1-c_4})^2\right)\sqrt{\kappa\widetilde{\kappa}} \qquad \text{(B.72)}
\end{aligned}$$

We add $T_1$ and $T_2$ and revisit equation (B.69):

$$\left\|(\mathbf{I}-\mathbf{A}_j^\top)^{-2}\mathbf{A}_j^\top\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}\right\|_{\mathbf{U}_j}\cdot\left\|\begin{bmatrix}\lambda_j^{1/2}\\0\end{bmatrix}\right\|_{\mathbf{U}_j}$$

$$
\begin{aligned}
&= T_1 + T_2 \\
&\leq \Bigg( \frac{2}{c_4} \cdot \Big(1 + (\frac{1+\sqrt{c_1c_4}}{1-c_4})^2\Big) + 3 \cdot \frac{1+\sqrt{c_1c_4}}{1-c_4} \cdot \frac{1+\sqrt{2}+\sqrt{c_4/c_1}}{c_4} \cdot (\sqrt{2}+\sqrt{c_4/c_1}) \Bigg)\sqrt{\kappa\widetilde{\kappa}}
\end{aligned} \tag{B.73}
$$

Then, we revisit equation (B.64):

$$
\begin{aligned}
&\left( \begin{bmatrix} \mathbf{H}^{1/2} \\ 0 \end{bmatrix}^\top \mathbf{\Phi}_\infty^{1/2} \right) \cdot \left( \mathbf{\Phi}_\infty^{1/2} (\mathbf{I} - \mathbf{A}^\top)^{-2} \mathbf{A}^\top \begin{bmatrix} \mathbf{H}^{1/2} \\ 0 \end{bmatrix} \right) \\
&+ \left( \mathbf{\Phi}_\infty^{1/2} \begin{bmatrix} \mathbf{H}^{1/2} \\ 0 \end{bmatrix} \right) \cdot \left( \begin{bmatrix} \mathbf{H}^{1/2} \\ 0 \end{bmatrix}^\top \mathbf{A}(\mathbf{I} - \mathbf{A})^{-2} \mathbf{\Phi}_\infty^{1/2} \right) \\
&\leq 2 \sum_{j=1}^{d} \left\| \begin{bmatrix} \lambda_j^{1/2} \\ 0 \end{bmatrix} \right\|_{(\mathbf{\Phi}_\infty)_j} \cdot \left\| (\mathbf{I} - \mathbf{A}_j^\top)^{-2} \mathbf{A}_j^\top \begin{bmatrix} \lambda_j^{1/2} \\ 0 \end{bmatrix} \right\|_{(\mathbf{\Phi}_\infty)_j} \\
&\leq 10\sigma^2 \sum_{j=1}^{d} \left\| \begin{bmatrix} \lambda_j^{1/2} \\ 0 \end{bmatrix} \right\|_{\mathbf{U}_j} \cdot \left\| (\mathbf{I} - \mathbf{A}_j^\top)^{-2} \mathbf{A}_j^\top \begin{bmatrix} \lambda_j^{1/2} \\ 0 \end{bmatrix} \right\|_{\mathbf{U}_j} \qquad \text{(using equation (B.61))} \\
&\leq 10\sigma^2 \cdot d \cdot \Bigg( \frac{2}{c_4} \cdot \Big(1 + (\frac{1+\sqrt{c_1c_4}}{1-c_4})^2\Big) \\
&\qquad + 3 \cdot \frac{1+\sqrt{c_1c_4}}{1-c_4} \cdot \frac{1+\sqrt{2}+\sqrt{c_4/c_1}}{c_4} \cdot (\sqrt{2}+\sqrt{c_4/c_1}) \Bigg)\sqrt{\kappa\widetilde{\kappa}} \\
&\leq C\sigma^2 d \sqrt{\kappa\widetilde{\kappa}}
\end{aligned} \tag{B.74}
$$

Where the equation in the penultimate line is obtained by summing over all eigen directions the bound implied by equation (B.73), and $C$ is a universal constant. □

**Lemma 44.**

$$
\begin{aligned}
\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \mathbb{E}\left[ \bar{\boldsymbol{\theta}}_{t,n}^{variance} \otimes \bar{\boldsymbol{\theta}}_{t,n}^{variance} \right] \rangle &\leq 5\frac{\sigma^2 d}{n-t} + C \cdot \frac{\sigma^2 d}{(n-t)^2} \cdot \sqrt{\kappa\widetilde{\kappa}} \\
&+ C \cdot \frac{\sigma^2 d}{n-t} (\kappa\widetilde{\kappa})^{11/4} \exp\left( - \frac{(n-t-1)c_2\sqrt{2c_1 - c_1^2}}{4\sqrt{\kappa\widetilde{\kappa}}} \right) \\
&+ C \cdot \frac{\sigma^2 d}{(n-t)^2} \cdot \exp\left( -(n+1)\frac{c_1 c_3^2}{\sqrt{\kappa\widetilde{\kappa}}} \right) \cdot (\kappa\widetilde{\kappa})^{7/2}\widetilde{\kappa}
\end{aligned}
$$

$$+ C \cdot \sigma^2 d \cdot (\kappa\widetilde{\kappa})^{7/4} \exp\left(-(n+1) \cdot \frac{c_2 c_3 \sqrt{2c_1 - c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\right)$$

*where, $C$ is a universal constant.*

*Proof.* We begin by recounting the expression for the covariance of the variance error of the tail-averaged iterate $\bar{\boldsymbol{\theta}}_{t,n}^{\text{variance}}$ from equation (B.15):

$$\begin{aligned}
\mathbb{E}\left[\bar{\boldsymbol{\theta}}_{t,n}^{\text{variance}} \otimes \bar{\boldsymbol{\theta}}_{t,n}^{\text{variance}}\right] = &\underbrace{\frac{1}{n-t}\big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^\top)^{-1}\mathcal{A}_{\mathcal{R}}^\top\big)(\mathcal{I} - \mathcal{B})^{-1}\widehat{\boldsymbol{\Sigma}}}_{\mathcal{E}_1 \overset{\text{def}}{=}} \\
&\underbrace{-\frac{1}{(n-t)^2}\big((\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-2}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^\top)^{-2}\mathcal{A}_{\mathcal{R}}^\top\big)(\mathcal{I} - \mathcal{B})^{-1}\widehat{\boldsymbol{\Sigma}}}_{\mathcal{E}_2 \overset{\text{def}}{=}} \\
&\underbrace{+\frac{1}{(n-t)^2}\big((\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-2}\mathcal{A}_{\mathcal{L}}^{n+1-t} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^\top)^{-2}(\mathcal{A}_{\mathcal{R}}^\top)^{n+1-t}\big)(\mathcal{I} - \mathcal{B})^{-1}\widehat{\boldsymbol{\Sigma}}}_{\mathcal{E}_3 \overset{\text{def}}{=}} \\
&\underbrace{-\frac{1}{(n-t)^2}\big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^\top)^{-1}\mathcal{A}_{\mathcal{R}}^\top\big)(\mathcal{I} - \mathcal{B})^{-2}(\mathcal{B}^{t+1} - \mathcal{B}^{n+1})\widehat{\boldsymbol{\Sigma}}}_{\mathcal{E}_4 \overset{\text{def}}{=}} \\
&\underbrace{+\frac{1}{(n-t)^2}\sum_{j=t+1}^{n}\big((\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}}^{n+1-j} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^\top)^{-1}(\mathcal{A}_{\mathcal{R}}^\top)^{n+1-j}\big)(\mathcal{I} - \mathcal{B})^{-1}\mathcal{B}^j\widehat{\boldsymbol{\Sigma}}}_{\mathcal{E}_5 \overset{\text{def}}{=}}
\end{aligned}$$

The goal is to bound $\langle \begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix}, \mathcal{E}_i\rangle$, for $i = 1, .., 5$.

For the case of $\mathcal{E}_1$, combining the fact that $\mathbb{E}\left[\boldsymbol{\theta}_\infty \otimes \boldsymbol{\theta}_\infty\right] = (\mathcal{I} - \mathcal{B})^{-1}\widehat{\boldsymbol{\Sigma}}$ and Lemma 41, we get:

$$\begin{aligned}
\langle \begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix}, \mathcal{E}_1\rangle &= \frac{1}{n-t}\langle \begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix}, \big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^\top)^{-1}\mathcal{A}_{\mathcal{R}}^\top\big)(\mathcal{I} - \mathcal{B})^{-1}\widehat{\boldsymbol{\Sigma}}\rangle \\
&= \frac{1}{n-t}\langle \begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix}, \big(\mathcal{I} + (\mathcal{I} - \mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I} - \mathcal{A}_{\mathcal{R}}^\top)^{-1}\mathcal{A}_{\mathcal{R}}^\top\big)\mathbb{E}\left[\boldsymbol{\theta}_\infty \otimes \boldsymbol{\theta}_\infty\right]\rangle \\
&\leq 5\frac{\sigma^2 d}{n-t} \qquad \text{(B.75)}
\end{aligned}$$

For the case of $\mathcal{E}_2$, we employ the result from Lemma 43, and this gives us:

$$\Big|\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \mathcal{E}_2 \rangle\Big| \leq \frac{C \cdot \sigma^2 d\sqrt{\kappa\widetilde{\kappa}}}{(n-t)^2} \tag{B.76}$$

For $i = 3$, we have:

$$\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \mathcal{E}_3 \rangle = \frac{1}{(n-t)^2} \langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \big((\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-2}\mathcal{A}_{\mathcal{L}}^{n+1-t} + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^\top)^{-2}(\mathcal{A}_{\mathcal{R}}^\top)^{n+1-t}\big)(\mathcal{I}-\mathcal{B})^{-1}\widehat{\mathbf{\Sigma}}\rangle$$

$$= \frac{1}{(n-t)^2}\Bigg(\langle (\mathbf{I}-\mathbf{A}^\top)^{-2}\mathbf{A}^\top \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{A}^{n-t}(\mathcal{I}-\mathcal{B})^{-1}\widehat{\mathbf{\Sigma}}\rangle$$

$$+ \langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix} \mathbf{A}(\mathbf{I}-\mathbf{A})^{-2}, (\mathcal{I}-\mathcal{B})^{-1}\widehat{\mathbf{\Sigma}}\ (\mathbf{A}^\top)^{n-t}\rangle\Bigg)$$

$$= \frac{4d}{(n-t)^2} \cdot \|(\mathbf{I}-\mathbf{A}^\top)^{-2}\mathbf{A}^\top \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}\| \cdot \|\mathbf{A}^{n-t}(\mathcal{I}-\mathcal{B})^{-1}\widehat{\mathbf{\Sigma}}\| \tag{B.77}$$

We will consider bounding $\|\mathbf{A}^{n-t}(\mathcal{I}-\mathcal{B})^{-1}\widehat{\mathbf{\Sigma}}\|$:

$$\begin{aligned}
\|\mathbf{A}^{n-t}(\mathcal{I}-\mathcal{B})^{-1}\widehat{\mathbf{\Sigma}}\| &\leq \sum_{i=0}^{\infty} \|\mathbf{A}^{n-t}\mathcal{B}^i\widehat{\mathbf{\Sigma}}\| \\
&\leq \frac{12\sqrt{2}}{\sqrt{1-\alpha^2}}\kappa(n-t)\alpha^{(n-t-1)/2}\Bigg(\sum_i \big(1 - \frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}\big)^i\Bigg)\|\widehat{\mathbf{\Sigma}}\| \\
&\qquad\qquad\text{(using Corollary 39)} \\
&= \frac{12\sqrt{2}}{\sqrt{1-\alpha^2}}\kappa(n-t)\alpha^{(n-t-1)/2} \cdot \frac{\sqrt{\kappa\widetilde{\kappa}}}{c_2c_3\sqrt{2c_1-c_1^2}} \cdot \|\widehat{\mathbf{\Sigma}}\| \\
&= \frac{12\sqrt{2}\sigma^2}{\sqrt{1-\alpha^2}}\kappa(n-t)\alpha^{(n-t-1)/2} \cdot \frac{\sqrt{\kappa\widetilde{\kappa}}}{c_2c_3\sqrt{2c_1-c_1^2}} \cdot (q+c\delta)^2\|\mathbf{H}\| \\
&\leq \frac{108\sqrt{2}\sigma^2}{\sqrt{1-\alpha^2}}\kappa(n-t)\alpha^{(n-t-1)/2} \cdot \frac{\sqrt{\kappa\widetilde{\kappa}}}{c_2c_3\sqrt{2c_1-c_1^2}} \cdot \delta^2\|\mathbf{H}\|
\end{aligned} \tag{B.78}$$

We also upper bound $\alpha$ as:

$$\alpha = 1 - \frac{c_2\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}} + c_2\sqrt{2c_1-c_1^2}}$$

$$
\begin{aligned}
&\leq 1 - \frac{c_2\sqrt{2c_1 - c_1^2}}{2\sqrt{\kappa\widetilde{\kappa}}} \\
&= e^{-\frac{c_2\sqrt{2c_1-c_1^2}}{2\sqrt{\kappa\widetilde{\kappa}}}} \qquad \text{(B.79)}
\end{aligned}
$$

Furthermore, for $\|(\mathbf{I} - \mathbf{A}^\top)^{-2}\mathbf{A}^\top \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix} \|$, we consider a bound in each eigendirection $j$ and accumulate the results subsequently:

$$
\begin{aligned}
&\|(\mathbf{I} - \mathbf{A}_j^\top)^{-2}\mathbf{A}_j^\top \begin{bmatrix} \lambda_j & 0 \\ 0 & 0 \end{bmatrix} \| \\
&\leq \frac{1}{(q-c\delta)^2} \cdot \frac{1-\delta\lambda_j}{\lambda_j} \cdot \sqrt{(1+c^2)(1-c)^2 + c^2\lambda_j^2(q^2+\delta^2)} \\
&\hspace{20em} \text{(using Lemma 33)} \\
&\leq \frac{\sqrt{7}}{(q-c\delta)^2} \cdot \frac{1}{\lambda_j} \\
&\leq \frac{\sqrt{7}}{(\gamma(1-\alpha))^2} \cdot \frac{1}{\lambda_j} \\
&\leq \frac{48(\kappa\widetilde{\kappa})^2}{(c_1c_4)^2}\frac{\mu^2}{\lambda_j} = \frac{48\widetilde{\kappa}^2}{(\delta c_4)^2}\frac{1}{\lambda_j}
\end{aligned}
$$

$$
\implies \|(\mathbf{I} - \mathbf{A}^\top)^{-2}\mathbf{A}^\top \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix} \| \leq \frac{48\widetilde{\kappa}^2}{(\delta c_4)^2} \cdot \frac{1}{\mu}
$$

Plugging this into equation B.77, we obtain:

$$
\begin{aligned}
\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \mathcal{E}_3 \rangle &\leq 41472\frac{\sigma^2 d}{n-t}(\kappa\widetilde{\kappa})^{11/4}\alpha^{(n-t-1)/2}\frac{1}{c_3c_4^2(c_1c_3)^{3/2}} \\
&\leq C\frac{\sigma^2 d}{n-t}(\kappa\widetilde{\kappa})^{11/4}\alpha^{(n-t-1)/2} \\
&\leq C\frac{\sigma^2 d}{n-t}(\kappa\widetilde{\kappa})^{11/4}\exp^{-\frac{(n-t-1)c_2\sqrt{2c_1-c_1^2}}{4\sqrt{\kappa\widetilde{\kappa}}}} \qquad \text{(B.80)}
\end{aligned}
$$

Next, let us consider $\mathcal{E}_4$:

$$
\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \mathcal{E}_4 \rangle
$$

$$= -\frac{1}{(n-t)^2}\langle \begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}, \big(\mathcal{I} + (\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}} + (\mathcal{I}-\mathcal{A}_{\mathcal{R}}^{\top})^{-1}\mathcal{A}_{\mathcal{R}}^{\top}\big)(\mathcal{I}-\mathcal{B})^{-2}(\mathcal{B}^{t+1}-\mathcal{B}^{n+1})\widehat{\mathbf{\Sigma}}\rangle$$

$$= -\frac{1}{(n-t)^2}\langle (\mathbf{I}-\mathbf{A}^{\top})^{-1}\begin{bmatrix} \mathbf{H} & 0 \\ 0 & 0 \end{bmatrix}(\mathbf{I}-\mathbf{A})^{-1}, (\mathcal{I}-\mathcal{D})(\mathcal{I}-\mathcal{B})^{-2}(\mathcal{B}^{t+1}-\mathcal{B}^{n+1})\widehat{\mathbf{\Sigma}}\rangle$$

$$= -\frac{1}{(q-c\delta)^2(n-t)^2}\langle \Bigg(\otimes_2 \begin{bmatrix} -(c\mathbf{I}-q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Bigg), (\mathcal{I}-\mathcal{B}+\mathcal{R})(\mathcal{I}-\mathcal{B})^{-2}(\mathcal{B}^{t+1}-\mathcal{B}^{n+1})\widehat{\mathbf{\Sigma}}\rangle$$

(using Lemma 32)

$$\le \frac{1}{(q-c\delta)^2(n-t)^2}\langle \Bigg(\otimes_2 \begin{bmatrix} -(c\mathbf{I}-q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Bigg), (\mathcal{I}-\mathcal{B}+\mathcal{R})(\mathcal{I}-\mathcal{B})^{-2}\mathcal{B}^{n+1}\widehat{\mathbf{\Sigma}}\rangle$$

$$\le \frac{1}{(q-c\delta)^2(n-t)^2}\cdot\Bigg(\langle \Bigg(\otimes_2 \begin{bmatrix} -(c\mathbf{I}-q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Bigg), (\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{n+1}\widehat{\mathbf{\Sigma}}\rangle$$
$$+ \langle \mathcal{R}^{\top}\Bigg(\otimes_2 \begin{bmatrix} -(c\mathbf{I}-q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Bigg), (\mathcal{I}-\mathcal{B})^{-2}\mathcal{B}^{n+1}\widehat{\mathbf{\Sigma}}\rangle\Bigg)$$

$$= \frac{1}{(q-c\delta)^2(n-t)^2}\cdot\Bigg(\langle \Bigg(\otimes_2 \begin{bmatrix} -(c\mathbf{I}-q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Bigg), (\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{n+1}\widehat{\mathbf{\Sigma}}\rangle$$
$$+ \langle \otimes_2 \begin{bmatrix} \delta \\ q \end{bmatrix} \otimes \Big(\mathcal{M} - \mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}}\Big)(\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1}(\mathbf{I}-\delta\mathbf{H}), (\mathcal{I}-\mathcal{B})^{-2}\mathcal{B}^{n+1}\widehat{\mathbf{\Sigma}}\rangle\Bigg)$$

$$\le \frac{1}{(q-c\delta)^2(n-t)^2}\cdot\Bigg(\langle \Bigg(\otimes_2 \begin{bmatrix} -(c\mathbf{I}-q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Bigg), (\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{n+1}\widehat{\mathbf{\Sigma}}\rangle$$
$$+ \widetilde{\kappa}\cdot\langle (\widehat{\mathbf{\Sigma}}/\sigma^2), (\mathcal{I}-\mathcal{B})^{-2}\mathcal{B}^{n+1}\widehat{\mathbf{\Sigma}}\rangle\Bigg) \qquad \text{(B.81)}$$

To bound $\| \otimes_2 \begin{bmatrix} -(c\mathbf{I}-q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix} \|$, we will consider a bound along each eigendirection and accumulate the results:

$$\| \otimes_2 \begin{bmatrix} -(c-q\lambda_j)\lambda_j^{-1/2} \\ (1-\delta\lambda_j)\lambda_j^{-1/2} \end{bmatrix} \| \le \frac{(c-q\lambda_j)^2 + (1-\delta\lambda_j)^2}{\lambda_j}$$

$$\leq 2 \cdot \frac{(1+c^2)+(q^2+\delta^2)\lambda_j^2}{\lambda_j}$$

$$\leq 2 \cdot \frac{2+5\delta^2\lambda_j^2}{\lambda_j} \leq \frac{14}{\lambda_j}$$

$$\implies \| \otimes_2 \begin{bmatrix} -(c\mathbf{I}-q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix} \| \leq \frac{14}{\mu}$$

Next, we bound $\|\mathcal{B}^k(\mathcal{I}-\mathcal{B})^{-1}\widehat{\mathbf{\Sigma}}\|$ (as a consequence of Lemma 37 with $\mathbf{Q}=\widehat{\mathbf{\Sigma}}$):

$$\begin{aligned}
\|\mathcal{B}^k(\mathcal{I}-\mathcal{B})^{-1}\widehat{\mathbf{\Sigma}}\| &\leq \frac{1}{\lambda_{\min}(\mathbf{G})}\|\mathbf{G}^\top\mathcal{B}^k(\mathcal{I}-\mathcal{B})^{-1}\widehat{\mathbf{\Sigma}}\| \\
&\leq \frac{1}{\lambda_{\min}(\mathbf{G})}\sum_{l=k}^{\infty}\|\mathbf{G}^\top\mathcal{B}^k\widehat{\mathbf{\Sigma}}\| \\
&\leq \frac{\sqrt{\kappa\widetilde{\kappa}}}{c_2c_3\sqrt{2c_1-c_1^2}}\kappa(\mathbf{G})\exp(-k\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}})\|\widehat{\mathbf{\Sigma}}\| \\
&\leq \frac{4\sigma^2\kappa}{\sqrt{1-\alpha^2}}\cdot\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_1c_3^2}\exp(-k\frac{c_1c_3^2}{\sqrt{\kappa\widetilde{\kappa}}})\cdot 9\delta^2\|\mathbf{H}\|_2 \\
&\leq \frac{36\sigma^2\kappa}{\sqrt{1-\alpha^2}}\cdot\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_1c_3^2}\exp(-k\frac{c_1c_3^2}{\sqrt{\kappa\widetilde{\kappa}}})\cdot\delta
\end{aligned}$$

This implies,

$$\langle\Bigg(\otimes_2\begin{bmatrix} -(c\mathbf{I}-q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Bigg),(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{n+1}\widehat{\mathbf{\Sigma}}\rangle \leq$$

$$504\cdot\frac{\kappa}{\sqrt{1-\alpha^2}}\cdot\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_1c_3^2}\exp\left(-(n+1)\frac{c_1c_3^2}{\sqrt{\kappa\widetilde{\kappa}}}\right)\cdot\frac{\delta}{\mu}\cdot\sigma^2 d$$

Furthermore,

$$\begin{aligned}
\frac{\widetilde{\kappa}}{\sigma^2}\langle\widehat{\mathbf{\Sigma}},(\mathcal{I}-\mathcal{B})^{-2}\mathcal{B}^{n+1}\widehat{\mathbf{\Sigma}}\rangle &= \frac{\widetilde{\kappa}}{\sigma^2}\langle(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{(n+1)/2}\widehat{\mathbf{\Sigma}},(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{(n+1)/2}\widehat{\mathbf{\Sigma}}\rangle \\
&\leq \frac{\widetilde{\kappa}}{\sigma^2}\|(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{(n+1)/2}\widehat{\mathbf{\Sigma}}\|^2\cdot d \\
&\leq 1296\frac{\sigma^2 d}{1-\alpha^2}\left(\kappa\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_1c_3^2}\right)^2\delta^2\widetilde{\kappa}\exp(-(n+1)\frac{c_1c_3^2}{\sqrt{\kappa\widetilde{\kappa}}})
\end{aligned}$$

This implies that,

$$\langle\Bigg(\otimes_2\begin{bmatrix} -(c\mathbf{I}-q\mathbf{H})\mathbf{H}^{-1/2} \\ (\mathbf{I}-\delta\mathbf{H})\mathbf{H}^{-1/2} \end{bmatrix}\Bigg),(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^{n+1}\widehat{\mathbf{\Sigma}}\rangle+\frac{\widetilde{\kappa}}{\sigma^2}\langle\widehat{\mathbf{\Sigma}},(\mathcal{I}-\mathcal{B})^{-2}\mathcal{B}^{n+1}\widehat{\mathbf{\Sigma}}\rangle$$

$$\leq 2592\frac{\sigma^2 d}{1-\alpha^2}\left(\kappa\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_1c_3^2}\right)^2\delta^2\widetilde{\kappa}\exp(-(n+1)\frac{c_1c_3^2}{\sqrt{\kappa\widetilde{\kappa}}})$$

$$\leq 2592\cdot\sigma^2 d\cdot\left(\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_1c_3^2}\right)^3\exp(-(n+1)\frac{c_1c_3^2}{\sqrt{\kappa\widetilde{\kappa}}})\cdot\delta^2\kappa^2\widetilde{\kappa} \tag{B.82}$$

Finally, we also note the following:

$$\frac{1}{(q-c\delta)}\leq\frac{1}{(\gamma(1-\alpha))}\leq\frac{\mu}{(1-\alpha)^2}\leq\frac{4\widetilde{\kappa}}{\delta c_4}$$

Plugging equation (B.82) into equation (B.81), we get:

$$\begin{aligned}\langle\begin{bmatrix}\mathbf{H}&0\\0&0\end{bmatrix},\mathcal{E}_4\rangle &= 2592\cdot\frac{\sigma^2 d}{(n-t)^2(q-c\delta)^2}\cdot\left(\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_1c_3^2}\right)^3\exp(-(n+1)\frac{c_1c_3^2}{\sqrt{\kappa\widetilde{\kappa}}})\cdot\delta^2\kappa^2\widetilde{\kappa}\\
&\leq 41472\cdot\frac{\sigma^2 d}{(n-t)^2}\cdot\frac{1}{c_4^2}\cdot\left(\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_1c_3^2}\right)^3\exp(-(n+1)\frac{c_1c_3^2}{\sqrt{\kappa\widetilde{\kappa}}})\cdot\kappa^2\widetilde{\kappa}^3\\
&= 41472\cdot\frac{\sigma^2 d}{(n-t)^2}\cdot\frac{1}{c_4^2(c_1c_3^2)^3}\cdot\exp\left(-(n+1)\frac{c_1c_3^2}{\sqrt{\kappa\widetilde{\kappa}}}\right)\cdot(\kappa\widetilde{\kappa})^{7/2}\widetilde{\kappa}\\
&\leq C\cdot\frac{\sigma^2 d}{(n-t)^2}\cdot\exp\left(-(n+1)\frac{c_1c_3^2}{\sqrt{\kappa\widetilde{\kappa}}}\right)\cdot(\kappa\widetilde{\kappa})^{7/2}\widetilde{\kappa}\end{aligned} \tag{B.83}$$

Next, we consider $\mathcal{E}_5$:

$$\begin{aligned}&\langle\begin{bmatrix}\mathbf{H}&0\\0&0\end{bmatrix},\mathcal{E}_5\rangle\\
&=\frac{1}{(n-t)^2}\sum_{j=t+1}^{n}\langle\begin{bmatrix}\mathbf{H}&0\\0&0\end{bmatrix},\big((\mathcal{I}-\mathcal{A}_{\mathcal{L}})^{-1}\mathcal{A}_{\mathcal{L}}^{n+1-j}+(\mathcal{I}-\mathcal{A}_{\mathcal{R}}^\top)^{-1}(\mathcal{A}_{\mathcal{R}}^\top)^{n+1-j}\big)(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^j\widehat{\mathbf{\Sigma}}\rangle\\
&=\frac{1}{(n-t)^2}\sum_{j=t+1}^{n}\Bigg(\langle(\mathcal{I}-\mathbf{A}^\top)^{-1}\mathbf{A}^\top\begin{bmatrix}\mathbf{H}&0\\0&0\end{bmatrix},\mathbf{A}^{n-j}(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^j\widehat{\mathbf{\Sigma}}\rangle\\
&\qquad+\langle\begin{bmatrix}\mathbf{H}&0\\0&0\end{bmatrix}\mathbf{A}(\mathcal{I}-\mathbf{A})^{-1},(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^j\widehat{\mathbf{\Sigma}}(\mathbf{A}^\top)^{n-j}\rangle\Bigg)\\
&\leq\frac{4d}{(n-t)^2}\sum_{j=t+1}^{n}\|(\mathbf{I}-\mathbf{A}^\top)^{-1}\mathbf{A}^\top\begin{bmatrix}\mathbf{H}&0\\0&0\end{bmatrix}\|\cdot\|\mathbf{A}^{n-j}(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^j\widehat{\mathbf{\Sigma}}\|\end{aligned} \tag{B.84}$$

In a manner similar to bounding $\|\mathbf{A}^{n-t}(\mathcal{I}-\mathcal{B})^{-1}\widehat{\mathbf{\Sigma}}\|$ as in equation (B.78), we can bound $\|\mathbf{A}^{n-j}(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^j\widehat{\mathbf{\Sigma}}\|$ as:

$$\|\mathbf{A}^{n-j}(\mathcal{I}-\mathcal{B})^{-1}\mathcal{B}^j\widehat{\mathbf{\Sigma}}\| \leq \frac{108\sqrt{2}\sigma^2}{\sqrt{1-\alpha^2}}\kappa(n-j)\alpha^{(n-j-1)/2}\cdot\frac{\sqrt{\kappa\widetilde{\kappa}}}{c_2c_3\sqrt{2c_1-c_1^2}}\cdot\exp^{-(\frac{jc_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}})}\cdot\delta^2\|\mathbf{H}\|$$

Furthermore, we will consider the bound $\|(\mathbf{I}-\mathbf{A}^\top)^{-1}\mathbf{A}^\top\begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix}\|$ along one eigen direction (by employing equation (B.17)) and collect the results:

$$\begin{aligned}\|(\mathbf{I}-\mathbf{A}_j^\top)^{-1}\mathbf{A}_j^\top\begin{bmatrix}\lambda_j & 0\\ 0 & 0\end{bmatrix}\| &\leq \frac{1+c^2}{q-c\delta}\leq\frac{2}{q-c\delta}\\ &\leq\frac{2}{\gamma(1-\alpha)}\leq\frac{4\widetilde{\kappa}}{\delta c_4}\\ \implies \|(\mathbf{I}-\mathbf{A}^\top)^{-1}\mathbf{A}^\top\begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix}\| &\leq\frac{4\widetilde{\kappa}}{\delta c_4}\end{aligned}$$

Plugging this into equation (B.84), and upper bounding the sum by $(n-t)$ times the largest term of the series:

$$\begin{aligned}\langle\begin{bmatrix}\mathbf{H} & 0\\ 0 & 0\end{bmatrix},\mathcal{E}_5\rangle &\leq 6912\cdot\sigma^2 d\cdot\frac{(\kappa\widetilde{\kappa})^{7/4}}{c_3c_4(c_1c_3)^{3/2}}\exp^{-(n+1)\cdot\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}}\\ &\leq C\cdot\sigma^2 d\cdot(\kappa\widetilde{\kappa})^{7/4}\cdot\exp^{-(n+1)\cdot\frac{c_2c_3\sqrt{2c_1-c_1^2}}{\sqrt{\kappa\widetilde{\kappa}}}}\end{aligned}\tag{B.85}$$

Summing up equations (B.75), (B.76), (B.80), (B.83), (B.85), the statement of the lemma follows. □

## *B.6 Proof of Theorem 9*

*Proof of Theorem 9.* The proof of the theorem follows through various lemmas that have been proven in the appendix:

- Section B.2 provides the bias-variance decomposition and provides an exact tensor expression governing the covariance of the bias error (through Lemma 11)and the variance error (Lemma 13).

- Section B.4 provides a scalar bound of the bias error through Lemma 40. The technical contribution of this section (which introduces a new potential function) is in Lemma 12.

- Section B.5 provides a scalar bound of the variance error through Lemma 44. The key technical contribution of this section is in the introduction of a stochastic process viewpoint of the proposed accelerated stochastic gradient method through Lemmas 14, 41. These lemmas provide a tight characterization of the stationary distribution of the covariance of the iterates of the accelerated method. Lemma 43 is necessary to show the sharp burn-in (up to log factors), beyond which the leading order term of the error is up to constants the statistically optimal error rate $\mathcal{O}(\sigma^2 d/n)$.

Combining the results of these lemmas, we obtain the following guarantee of Algorithm 3:

$$\begin{aligned}\mathbb{E}\left[P(\bar{\mathbf{x}}_{t,n})\right] - P(\mathbf{x}^*) \leq C \cdot \frac{(\kappa\widetilde{\kappa})^{9/4} d\kappa}{(n-t)^2} \cdot \exp\left(-\frac{t+1}{9\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \big(P(\mathbf{x}_0) - P(\mathbf{x}^*)\big) \\ + C \cdot (\kappa\widetilde{\kappa})^{5/4} d\kappa \cdot \exp\left(\frac{-n}{9\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \big(P(\mathbf{x}_0) - P(\mathbf{x}^*)\big) + 5\frac{\sigma^2 d}{n-t} \\ + C \cdot \frac{\sigma^2 d}{(n-t)^2}\sqrt{\kappa\widetilde{\kappa}} + C \cdot \sigma^2 d \cdot (\kappa\widetilde{\kappa})^{7/4} \cdot \exp\left(\frac{-(n+1)}{9\sqrt{\kappa\widetilde{\kappa}}}\right) \\ + C \cdot \frac{\sigma^2 d}{n-t} (\kappa\widetilde{\kappa})^{11/4} \exp\left(-\frac{(n-t-1)}{30\sqrt{\kappa\widetilde{\kappa}}}\right) \\ + C \cdot \frac{\sigma^2 d}{(n-t)^2} \cdot \exp\left(-\frac{(n+1)}{9\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot (\kappa\widetilde{\kappa})^{7/2}\widetilde{\kappa}\end{aligned}$$

Where, $C$ is a universal constant.

$\square$

# Appendix C

# APPENDIX: UNDERSTANDING THE BEHAVIOR OF MOMENTUM WITH STOCHASTIC GRADIENTS

## *C.1 Suboptimality of HB: Proof of Proposition 17*

Before proceeding to the proof, we introduce some additional notation. Let $\boldsymbol{\theta}_{t+1}^{(j)}$ denote the concatenated and centered estimates in the $j^{\text{th}}$ direction for $j = 1, 2$.

$$\boldsymbol{\theta}_{t+1}^{(j)} \overset{\text{def}}{=} \begin{bmatrix} \mathbf{w}_{t+1}^{(j)} - (\mathbf{w}^*)^{(j)} \\ \mathbf{w}_{t}^{(j)} - (\mathbf{w}^*)^{(j)} \end{bmatrix}, \quad j = 1, 2.$$

Since the distribution over $x$ is such that the coordinates are decoupled, we see that $\boldsymbol{\theta}_{t+1}^{(j)}$ can be written in terms of $\boldsymbol{\theta}_{t}^{(j)}$ as:

$$\boldsymbol{\theta}_{t+1}^{(j)} = \widehat{\mathbf{A}}_{t+1}^{(j)} \boldsymbol{\theta}_{t}^{(j)}, \text{ with } \widehat{\mathbf{A}}_{t+1}^{(j)} = \begin{bmatrix} 1 + \alpha - \delta(a_{t+1}^{(j)})^2 & -\alpha \\ 1 & 0 \end{bmatrix}.$$

Let $\boldsymbol{\Phi}_{t+1}^{(j)} \overset{\text{def}}{=} \mathbb{E}\left[\boldsymbol{\theta}_{t+1}^{(j)} \otimes \boldsymbol{\theta}_{t+1}^{(j)}\right]$ denote the covariance matrix of $\boldsymbol{\theta}_{t+1}^{(j)}$. We have $\boldsymbol{\Phi}_{t+1}^{(j)} = \mathcal{B}^{(j)} \boldsymbol{\Phi}_{t}^{(j)}$ with, $\mathcal{B}^{(j)}$ defined as

$$\mathcal{B}^{(j)} \overset{\text{def}}{=} \begin{bmatrix} \mathbb{E}\left[(1+\alpha-\delta(a^{(j)})^2)^2\right] & \mathbb{E}\left[-\alpha(1+\alpha-\delta(a^{(j)})^2)\right] & \mathbb{E}\left[-\alpha(1+\alpha-\delta(a^{(j)})^2\right] & \alpha^2 \\ \mathbb{E}\left[(1+\alpha-\delta(a^{(j)})^2)\right] & 0 & -\alpha & 0 \\ \mathbb{E}\left[(1+\alpha-\delta(a^{(j)})^2)\right] & -\alpha & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} (1+\alpha-\delta\sigma_j^2)^2 + (c-1)(\delta\sigma_j^2)^2 & -\alpha(1+\alpha-\delta\sigma_j^2) & -\alpha(1+\alpha-\delta\sigma_j^2) & \alpha^2 \\ (1+\alpha-\delta\sigma_j^2) & 0 & -\alpha & 0 \\ (1+\alpha-\delta\sigma_j^2) & -\alpha & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

We prove Proposition 17 by showing that for any choice of stepsize and momentum, either of the two holds:

- $\mathcal{B}^{(1)}$ has an eigenvalue larger than 1, or,
- the largest eigenvalue of $\mathcal{B}^{(2)}$ is greater than $1 - \frac{500}{\kappa}$.

This is formalized in the following two lemmas.

**Lemma 45.** *If the stepsize $\delta$ is such that $\delta\sigma_1^2 \geq \frac{2\left(1-\alpha^2\right)}{c+(c-2)\alpha}$, then $\mathcal{B}^{(1)}$ has an eigenvalue $\geq 1$.*

**Lemma 46.** *If the stepsize $\delta$ is such that $\delta\sigma_1^2 < \frac{2\left(1-\alpha^2\right)}{c+(c-2)\alpha}$, then $\mathcal{B}^{(2)}$ has an eigenvalue of magnitude $\geq 1 - \frac{500}{\kappa}$.*

Given this notation, we can now consider the $j^{th}$ dimension without the superscripts; when needed, they will be made clear in the exposition. Denoting $x \overset{\text{def}}{=} \delta\sigma^2$ and $t \overset{\text{def}}{=} 1+\alpha-x$, we have:

$$\mathcal{B} = \begin{bmatrix} t^2+(c-1)x^2 & -\alpha t & -\alpha t & \alpha^2 \\ t & 0 & -\alpha & 0 \\ t & -\alpha & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

## C.1.1 Proof

The analysis goes via computation of the characteristic polynomial of $\mathcal{B}$ and evaluating it at different values to obtain bounds on its roots.

**Lemma 47.** *The characteristic polynomial of $\mathcal{B}$ is:*

$$D(z) = z^4 - (t^2+(c-1)x^2)z^3 + (2\alpha t^2 - 2\alpha^2)z^2 + (-t^2+(c-1)x^2)\alpha^2 z + \alpha^4.$$

*Proof.* We first begin by writing out the expression for the determinant:

$$Det(\mathcal{B}-z\mathcal{I}) = \begin{vmatrix} t^2+(c-1)x^2-z & -\alpha t & -\alpha t & \alpha^2 \\ t & -z & -\alpha & 0 \\ t & -\alpha & -z & 0 \\ 1 & 0 & 0 & -z \end{vmatrix}.$$

expanding along the first column, we have:

$$\begin{aligned} Det(\mathcal{B}-z\mathcal{I}) &= (t^2+(c-1)x^2-z)(\alpha^2 z - z^3) - t(-\alpha t z^2 + \alpha^2 t z) + t(-\alpha t(\alpha z) + z\cdot \alpha t z) - (z\cdot \alpha^2 z - \alpha^4) \\ &= (t^2+(c-1)x^2-z)(\alpha^2 z - z^3) - 2t(\alpha^2 t z - \alpha t z^2) - (\alpha^2 z^2 - \alpha^4). \end{aligned}$$

Expanding the terms yields the expression in the lemma. □

The next corollary follows by some simple arithmetic manipulations.

**Corollary 48.** *Substituting $z = 1-\tau$ in the characteristic equation of Lemma 47, we have:*

$$\begin{aligned} D(1-\tau) &= \tau^4 + \tau^3(-4+t^2+(c-1)x^2) + \tau^2(6-3t^2-3(c-1)x^2-2\alpha^2+2\alpha t^2) \\ &\quad + \tau(-4+3t^2+3(c-1)x^2+4\alpha^2-4\alpha t^2-(c-1)x^2\alpha^2+t^2\alpha^2) \\ &\quad + (1-t^2-(c-1)x^2-2\alpha^2+2\alpha t^2+(c-1)x^2\alpha^2-t^2\alpha^2+\alpha^4) \\ &= \tau^4+\tau^3[-(3+\alpha)(1-\alpha)-2x(1+\alpha)+cx^2] \\ &\quad + \tau^2[(3-4\alpha-\alpha^2+2\alpha^3)-2x(1+\alpha)(2\alpha-3)+x^2(2\alpha-3c)] \\ &\quad + \tau[-(1-\alpha)^2(1-\alpha^2)-2x(3-\alpha)(1-\alpha^2)+x^2(3c-4\alpha+(2-c)\alpha^2)] \\ &\quad + x(1-\alpha)[2(1-\alpha^2)-x(c+(c-2)\alpha)]. \end{aligned} \tag{C.1}$$

*Proof of Lemma 45.* The first observation necessary to prove the lemma is that the characteristic polynomial $D(z)$ approaches $\infty$ as $z \to \infty$, i.e., $\lim_{z\to\infty} D(z) = +\infty$.

Next, we evaluate the characteristic polynomial at 1, i.e. compute $D(1)$. This follows in a straightforward manner from Corollary 48 by substituting $\tau = 0$ in equation (C.1), and this yields,

$$D(1) = (1-\alpha)x\cdot\Big(2(1-\alpha^2) - x(1-\alpha) - (c-1)x(1+\alpha)\Big).$$

As $\alpha < 1$, $x = \delta\sigma^2 > 0$, we have the following by setting $D(1) \leq 0$ and solving for $x$:

$$x \geq \frac{2(1-\alpha^2)}{c + (c-2)\alpha}.$$

Since $D(1) \leq 0$ and $D(z) \geq 0$ as $z \to \infty$, there exists a root of $D(\cdot)$ which is $\geq 1$. □

**Remark 49.** *The above characterization is striking in the sense that for any $c > 1$, increasing the momentum parameter $\alpha$ naturally requires the reduction in the step size $\delta$ to permit the convergence of the algorithm, which is not observed when fast gradient methods are employed in deterministic optimization. For instance, in the case of deterministic optimization, setting $c = 1$ yields $\delta\sigma_1^2 < 2(1+\alpha)$. On the other hand, when employing the stochastic heavy ball method with $x^{(j)} = 2\sigma_j^2$, we have the condition that $c = 2$, and this implies, $\delta\sigma_1^2 < \frac{2(1-\alpha^2)}{2} = 1 - \alpha^2$.*

We now prove Lemma 46. We first consider the large momentum setting.

**Lemma 50.** *When the momentum parameter $\alpha$ is set such that $1 - 450/\kappa \leq \alpha \leq 1$, $\mathcal{B}$ has an eigenvalue of magnitude $\geq 1 - \frac{450}{\kappa}$.*

*Proof.* This follows easily from the fact that $\det(\mathcal{B}) = \alpha^4 = \prod_{j=1}^{4} \lambda_j(\mathcal{B}) \leq (\lambda_{\max}(\mathcal{B}))^4$, thus implying $1 - 450/\kappa \leq \alpha \leq |\lambda_{\max}(\mathcal{B})|$. □

**Remark 51.** *Note that the above lemma holds for any value of the learning rate $\delta$, and holds for every eigen direction of $\mathbf{H}$. Thus, for "large" values of momentum, the behavior of stochastic heavy ball does degenerate to the behavior of stochastic gradient descent.*

We now consider the setting where momentum is bounded away from 1.

**Corollary 52.** *Consider $\mathcal{B}^{(2)}$, by substituting $\tau = l/\kappa$, $x = \delta\lambda_{\min} = c(\delta\sigma_1^2)/\kappa$ in equation* (C.1) *and accumulating terms in varying powers of $1/\kappa$, we obtain:*

$$\begin{aligned} G(l) &\overset{def}{=} \frac{c^3(\delta\sigma_1^2)^2 l^3}{\kappa^5} + \frac{l^4 - 2c(\delta\sigma_1^2)l^3(1+\alpha) + (2\alpha - 3c)c^2(\delta\sigma_1^2)^2 l^2}{\kappa^4} \\ &+ \frac{-(3+\alpha)(1-\alpha)l^3 - 2(1+\alpha)(2\alpha-3)c(\delta\sigma_1^2)l^2 + (3c - 4\alpha + (2-c)\alpha^2)c^2(\delta\sigma_1^2)^2 l}{\kappa^3} \end{aligned}$$

$$+\frac{(3-4\alpha-\alpha^2+2\alpha^3)l^2-2c(\delta\sigma_1^2)l(3-\alpha)(1-\alpha^2)-c^2(\delta\sigma_1^2)^2(1-\alpha)(c+(c-2)\alpha)}{\kappa^2}$$
$$+\frac{-(1-\alpha)^2(1-\alpha^2)l+2c(\delta\sigma_1^2)(1-\alpha)(1-\alpha^2)}{\kappa} \tag{C.2}$$

**Lemma 53.** *Let $2 < c < 3000$, $0 \le \alpha \le 1-\frac{450}{\kappa}$, $l = 1+\frac{2c(\delta\sigma_1^2)}{1-\alpha}$. Then, $G(l) \le 0$.*

*Proof.* Since $(\delta\sigma_1^2) \le \frac{2(1-\alpha^2)}{c+(c-2)\alpha}$, this implies $\frac{(\delta\sigma_1^2)}{1-\alpha} \le \frac{2(1+\alpha)}{c+(c-2)\alpha} \le \frac{4}{c}$, thus implying, $1 \le l \le 9$.

Substituting the value of $l$ in equation (C.2), the coefficient of $\mathcal{O}(1/\kappa)$ is $-(1-\alpha)^3(1+\alpha)$.

We will bound this term along with $(3-4\alpha-\alpha^2+2\alpha^3)l^2/\kappa^2 = (1-\alpha)^2(3+2\alpha)l^2/\kappa^2$ to obtain:

$$\begin{aligned}\frac{-(1-\alpha)^3(1+\alpha)}{\kappa}+\frac{(1-\alpha)^2(3+2\alpha)l^2}{\kappa^2} &\le \frac{-(1-\alpha)^3(1+\alpha)}{\kappa}+\frac{405(1-\alpha)^2}{\kappa^2}\\ &\le \frac{(1-\alpha)^2}{\kappa}\Big(\frac{405}{\kappa}-(1-\alpha^2)\Big)\\ &\le \frac{(1-\alpha)^2}{\kappa}\Big(\frac{405}{\kappa}-(1-\alpha)\Big) \le -\frac{45\cdot 450^2}{\kappa^4},\end{aligned}$$

where, we use the fact that $\alpha < 1$, $l \le 9$. The natural implication of this bound is that the terms that are lower order, such as $\mathcal{O}(1/\kappa^4)$ and $\mathcal{O}(1/\kappa^5)$ will be negative owing to the large constant above. Let us verify that this is indeed the case by considering the terms having powers of $\mathcal{O}(1/\kappa^4)$ and $\mathcal{O}(1/\kappa^5)$ from equation (C.2):

$$\begin{aligned}&\frac{c^3(\delta\sigma_1^2)^2l^3}{\kappa^5}+\frac{l^4-2c(\delta\sigma_1^2)l^3(1+\alpha)+(2\alpha-3c)c^2(\delta\sigma_1^2)^2l^2}{\kappa^4}-\frac{45\cdot 450^2}{\kappa^4}\\ &\le \frac{c^3(\delta\sigma_1^2)^2l^3}{\kappa^5}+\frac{l^4}{\kappa^4}-\frac{45\cdot 450^2}{\kappa^4}\\ &\le \frac{cl^3}{\kappa^5}+\frac{(9^4-(45\cdot 450^2))}{\kappa^4} \le \frac{9^3c+9^4-(45\cdot 450^2)}{\kappa^4}\end{aligned}$$

The expression above evaluates to $\le 0$ given an upperbound on the value of $c$. The expression above follows from the fact that $l \le 9, \kappa \ge 1$.

Next, consider the terms involving $\mathcal{O}(1/\kappa^3)$ and $\mathcal{O}(1/\kappa^2)$, in particular,

$$\begin{aligned}&\frac{(3c-4\alpha+(2-c)\alpha^2)c^2(\delta\sigma_1^2)^2l}{\kappa^3}-\frac{c^2(\delta\sigma_1^2)^2(1-\alpha)(c+(c-2)\alpha)}{\kappa^2}\\ &\le \frac{c^2(\delta\sigma_1^2)^2}{\kappa^2}\big(\frac{l(3c+2)}{\kappa}-(1-\alpha)(c+(c-2)\alpha)\big)\end{aligned}$$

$$\leq \frac{c^2(\delta\sigma_1^2)^2}{\kappa^2}\big(\frac{5cl}{\kappa} - (1-\alpha)(c+(c-2)\alpha)\big)$$
$$\leq \frac{c^2(\delta\sigma_1^2)^2}{\kappa^2}\big(\frac{5cl}{\kappa} - (1-\alpha)c\big)$$
$$\leq \frac{c^3(\delta\sigma_1^2)^2}{\kappa^2}\big(\frac{5l}{\kappa} - \frac{450}{\kappa}\big)$$
$$\leq \frac{c^3(\delta\sigma_1^2)^2}{\kappa^2}\cdot\frac{-405}{\kappa} \leq 0.$$

Next,

$$\frac{-2(1+\alpha)(2\alpha-3)c(\delta\sigma_1^2)l^2}{\kappa^3} - \frac{2c(\delta\sigma_1^2)l(3-\alpha)(1-\alpha^2)}{\kappa^2}$$
$$\leq \frac{2(1+\alpha)c(\delta\sigma_1^2)l}{\kappa^2}\left(\frac{-(2\alpha-3)l}{\kappa} - (3-\alpha)(1-\alpha)\right)$$
$$\leq \frac{2(1+\alpha)c(\delta\sigma_1^2)l}{\kappa^2}\left(\frac{3l}{\kappa} - 2(1-\alpha)\right)$$
$$\leq \frac{2(1+\alpha)c(\delta\sigma_1^2)l}{\kappa^2}\left(\frac{3l}{\kappa} - \frac{2\cdot 450}{\kappa}\right)$$
$$\leq \frac{2(1+\alpha)c(\delta\sigma_1^2)l}{\kappa^2}\left(\frac{3\cdot 27}{\kappa} - \frac{2\cdot 450}{\kappa}\right) \leq 0.$$

In both these cases, we used the fact that $\alpha \leq 1 - \frac{450}{\kappa}$ implying $-(1-\alpha) \leq \frac{-450}{\kappa}$. Finally, other remaining terms are negative. □

Before rounding up the proof of the proposition, we need the following lemma to ensure that our lower bounds on the largest eigenvalue of $\mathcal{B}$ indeed affect the algorithm's rates and are true *irrespective* of where the algorithm is begun. Note that this allows our result to be *much stronger* than typical optimization lowerbounds that rely on specific initializations to ensure a component along the largest eigendirection of the update operator, for which bounds are proven.

**Lemma 54.** *For any starting iterate* $\mathbf{w}_0 \neq \mathbf{w}^*$*, the HB method produces a non-zero component along the largest eigen direction of* $\mathcal{B}$*.*

*Proof.* We note that in a similar manner as other proofs, it suffices to argue for each dimension of the problem separately. But before we start looking at each dimension separately, let us consider the $j^{\text{th}}$ dimension, and detail the approach we use to prove the claim: the idea is to examine the subspace spanned by covariance $\mathbb{E}\left[\boldsymbol{\theta}_{\cdot}^{(j)} \otimes \boldsymbol{\theta}_{\cdot}^{(j)}\right]$ of the iterates $\boldsymbol{\theta}_0^{(j)}, \boldsymbol{\theta}_1^{(j)}, \boldsymbol{\theta}_2^{(j)}, ...,$ for every starting iterate $\boldsymbol{\theta}_0^{(j)} \neq \begin{bmatrix}0, 0\end{bmatrix}^\top$ and prove that the largest eigenvector of the expected operator $\mathcal{B}^{(j)}$ is not orthogonal to this subspace. This implies that there exists a non-zero component of $\mathbb{E}\left[\boldsymbol{\theta}_{\cdot}^{(j)} \otimes \boldsymbol{\theta}_{\cdot}^{(j)}\right]$ in the largest eigen direction of $\mathcal{B}^{(j)}$, and this decays at a rate that is at best $\lambda_{\max}(\mathcal{B}^{(j)})$.

Since $\mathcal{B}^{(j)} \in \mathbb{R}^{4\times 4}$, we begin by examining the expected covariance spanned by the iterates $\boldsymbol{\theta}_0^{(j)}, \boldsymbol{\theta}_1^{(j)}, \boldsymbol{\theta}_2^{(j)}, \boldsymbol{\theta}_3^{(j)}$. Let $\mathbf{w}_0^{(j)} - (\mathbf{w}^*)^{(j)} = \mathbf{w}_{-1}^{(j)} - (\mathbf{w}^*)^{(j)} = k^{(j)}$. Now, this implies $\boldsymbol{\theta}_0^{(j)} = k^{(j)} \cdot \begin{bmatrix}1, 1\end{bmatrix}^\top$. Then,

$$\boldsymbol{\theta}_1^{(j)} = k^{(j)} \widehat{\mathbf{A}}_1^{(j)} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \text{with } \widehat{\mathbf{A}}_1^{(j)} = \begin{bmatrix} 1 + \alpha - \delta \widehat{\mathbf{H}}_1^{(j)} & -\alpha \\ 1 & 0 \end{bmatrix}, \text{ where } \widehat{\mathbf{H}}_1^{(j)} = (a_1^{(j)})^2.$$

This implies that $k$ just appears as a scale factor. This in turn implies that in order to analyze the subspace spanned by the covariance of iterates $\boldsymbol{\theta}_0^{(j)}, \boldsymbol{\theta}_1^{(j)}, ...,$ we can assume $k^{(j)} = 1$ without any loss in generality. This implies, $\boldsymbol{\theta}_0^{(j)} = \begin{bmatrix}1, 1\end{bmatrix}^\top$. Note that with this in place, we see that we can now drop the superscript $j$ that represents the dimension, since the analysis decouples across the dimensions $j \in \{1, 2\}$. Furthermore, let the entries of the vector $\boldsymbol{\theta}_k$ be represented as $\boldsymbol{\theta}_k \overset{\text{def}}{=} \begin{bmatrix}\theta_{k1} & \theta_{k2}\end{bmatrix}^\top$ Next, denote $1 + \alpha - \delta \widehat{\mathbf{H}}_k = \hat{t}_k$. This implies,

$$\widehat{\mathbf{A}}_k = \begin{bmatrix} \hat{t}_k & -\alpha \\ 1 & 0 \end{bmatrix}.$$

Furthermore,

$$\boldsymbol{\theta}_1 = \widehat{\mathbf{A}}_1 \boldsymbol{\theta}_0 = \begin{bmatrix} \hat{t}_1 - \alpha \\ 1 \end{bmatrix}, \ \boldsymbol{\theta}_2 = \widehat{\mathbf{A}}_2 \boldsymbol{\theta}_1 = \begin{bmatrix} \hat{t}_2(\hat{t}_1 - \alpha) - \alpha \\ \hat{t}_1 - \alpha \end{bmatrix},$$
$$\boldsymbol{\theta}_3 = \widehat{\mathbf{A}}_3 \boldsymbol{\theta}_2 = \begin{bmatrix} \hat{t}_3(\hat{t}_2(\hat{t}_1 - \alpha) - \alpha) - \alpha(\hat{t}_1 - \alpha) \\ \hat{t}_2(\hat{t}_1 - \alpha) - \alpha \end{bmatrix}. \tag{C.3}$$

Let us consider the vectorized form of $\boldsymbol{\Phi}_j = \mathbb{E}\left[\boldsymbol{\theta}_j \otimes \boldsymbol{\theta}_j\right]$, and we denote this as $\text{vec}(\boldsymbol{\Phi}_j)$. Note that $\text{vec}(\boldsymbol{\Phi}_j)$ makes $\boldsymbol{\Phi}_j$ become a column vector of size $4 \times 1$. Now, consider $\text{vec}(\boldsymbol{\Phi}_j)$ for

$j = 0, 1, 2, 3$ and concatenate these to form a matrix that we denote as $\mathcal{D}$, i.e.

$$\mathcal{D} = \begin{bmatrix} \text{vec}(\boldsymbol{\Phi}_0) & \text{vec}(\boldsymbol{\Phi}_1) & \text{vec}(\boldsymbol{\Phi}_2) & \text{vec}(\boldsymbol{\Phi}_3) \end{bmatrix}.$$

Now, since we note that $\boldsymbol{\Phi}_j$ is a symmetric $2 \times 2$ matrix, $\mathcal{D}$ should contain two identical rows implying that it has an eigenvalue that is zero and a corresponding eigenvector that is $\begin{bmatrix} 0 & -1/\sqrt{2} & 1/\sqrt{(2)} & 0 \end{bmatrix}^\top$. It turns out that this is also an eigenvector of $\mathcal{B}$ with an eigenvalue $\alpha$. Note that $\det(\mathcal{B}) = \alpha^4$. This implies there are two cases that we need to consider: (i) when all eigenvalues of $\mathcal{B}$ have the same magnitude ($= \alpha$). In this case, we are already done, because there exists at least one non zero eigenvalue of $\mathcal{D}$ and this should have some component along one of the eigenvectors of $\mathcal{B}$ and we know that all eigenvectors have eigenvalues with a magnitude equal to $\lambda_{\max}(\mathcal{B})$. Thus, there exists an iterate which has a non-zero component along the largest eigendirection of $\mathcal{B}$. (ii) the second case is the situation when we have eigenvalues with different magnitudes. In this case, note that $\det(\mathcal{B}) = \alpha^4 < (\lambda_{\max}(\mathcal{B}))^4$ implying $\lambda_{\max}(\mathcal{B}) > \alpha$. In this case, we need to prove that $\mathcal{D}$ spans a three-dimensional subspace; if it does, it contains a component along the largest eigendirection of $\mathcal{B}$ which will round up the proof. Since we need to understand whether $\mathcal{D}$ spans a three dimensional subspace, we can consider a different (yet related) matrix, which we call $\mathcal{R}$ and this is defined as:

$$\mathcal{R} \overset{\text{def}}{=} \mathbb{E}\left( \begin{bmatrix} \theta_{01}^2 & \theta_{11}^2 & \theta_{21}^2 \\ \theta_{01}\theta_{02} & \theta_{11}\theta_{12} & \theta_{21}\theta_{22} \\ \theta_{02}^2 & \theta_{12}^2 & \theta_{22}^2 \end{bmatrix} \right)$$

Given the expressions for $\{\boldsymbol{\theta}_j\}_{j=0}^3$ (by definition of $\boldsymbol{\theta}_0$ and using equation (C.3)), we can substitute to see that $\mathcal{R}$ has the following expression:

$$\mathcal{R} = \begin{bmatrix} 1 & \mathbb{E}\left[(\hat{t}_1 - \alpha)^2\right] & \mathbb{E}\left[(\hat{t}_2(\hat{t}_1 - \alpha) - \alpha)^2\right] \\ 1 & \mathbb{E}\left[\hat{t}_1 - \alpha\right] & \mathbb{E}\left[((\hat{t}_2(\hat{t}_1 - \alpha) - \alpha))(\hat{t}_1 - \alpha)\right] \\ 1 & 1 & \mathbb{E}\left[(\hat{t}_1 - \alpha)^2\right] \end{bmatrix}.$$

If we compute and prove that $\det(\mathcal{R}) \neq 0$, we are done since that implies that $\mathcal{R}$ has three non-zero eigenvalues.

This implies, we first define the following: let $q_\gamma = (t-\gamma)^2 + (c-1)x^2$. Then, $\mathcal{R}$ can be expressed as:

$$\begin{aligned}
\det(\mathcal{R}) &= \det\left( \begin{bmatrix} 1 & q_\alpha & q_0 q_\alpha - 2\alpha t(t-\alpha) + \alpha^2 \\ 1 & t-\alpha & tq_\alpha - \alpha(t-\alpha) \\ 1 & 1 & q_\alpha \end{bmatrix} \right) \\
&= \det\left( \begin{bmatrix} 1 & q_\alpha & q_\alpha(q_0 - q_\alpha) - 2\alpha t(t-\alpha) + \alpha^2 \\ 1 & t-\alpha & tq_\alpha - \alpha(t-\alpha) - (t-\alpha)q_\alpha \\ 1 & 1 & 0 \end{bmatrix} \right) \\
&= \det\left( \begin{bmatrix} 1 & q_\alpha - 1 & q_\alpha(q_0 - q_\alpha) - 2\alpha t(t-\alpha) + \alpha^2 \\ 1 & t-\alpha-1 & tq_\alpha - \alpha(t-\alpha) - (t-\alpha)q_\alpha \\ 1 & 0 & 0 \end{bmatrix} \right) \\
&= \det\left( \begin{bmatrix} 0 & q_\alpha - 1 & q_\alpha(q_0 - q_\alpha) - 2\alpha t(t-\alpha) + \alpha^2 \\ 0 & t-\alpha-1 & tq_\alpha - \alpha(t-\alpha) - (t-\alpha)q_\alpha \\ 1 & 0 & 0 \end{bmatrix} \right)
\end{aligned}$$

Note: (i) $q_\alpha - 1 = (t-\alpha)^2 - 1 + (c-1)x^2 = (1-x)^2 - 1 + (c-1)x^2 = -2x + x^2 + (c-1)x^2 = -2x + cx^2$.

(ii) $t - \alpha - 1 = -x$

(iii) $\alpha(q_\alpha - (t-\alpha)) = \alpha((t-\alpha)^2 - (t-\alpha) + (c-1)x^2) = \alpha((1-x)(-x) + (c-1)x^2) = \alpha x(-1+cx)$

(iv) $q_0 - q_\alpha = t^2 - (t-\alpha)^2 = \alpha(2t-\alpha) = 2t\alpha - \alpha^2$.

Then,

$$\begin{aligned}
(2\alpha t - \alpha^2)q_\alpha - 2\alpha t(t-\alpha) + \alpha^2 &= 2t\alpha(q_\alpha - (t-\alpha)) + \alpha^2(1 - q_\alpha) \\
&= 2t\alpha(-x + cx^2) - \alpha^2(-2x + cx^2) \\
&= -2t\alpha x + 2x\alpha^2 + 2t\alpha c x^2 - c\alpha^2 x^2 \\
&= 2\alpha x(-t + \alpha) + c\alpha x^2(2t - \alpha) \\
&= -2\alpha x(1-x) + 2c\alpha x^2(1-x) + c\alpha^2 x^2 \\
&= 2\alpha x(1-x)(-1+cx) + c\alpha^2 x^2.
\end{aligned}$$

Then,

$$\begin{aligned}
\det(\mathcal{R}) &= \det\left(\begin{bmatrix} 0 & x(cx-2) & 2\alpha x(1-x)(-1+cx) + c\alpha^2 x^2 \\ 0 & -x & \alpha x(cx-1) \\ 1 & 0 & 0 \end{bmatrix}\right) \\
&= x^2\alpha\det\left(\begin{bmatrix} 0 & (cx-2) & c\alpha x + 2(1-x)(cx-1) \\ 0 & -1 & cx-1 \\ 1 & 0 & 0 \end{bmatrix}\right) \\
&= x^3\alpha\det\left(\begin{bmatrix} 0 & c & c\alpha - 2(cx-1) \\ 0 & -1 & cx-1 \\ 1 & 0 & 0 \end{bmatrix}\right)
\end{aligned}$$

Then,

$$\begin{aligned}
\det(\mathcal{R}) &= x^3\alpha\Big(c(-1+cx) - 2(-1+cx) + c\alpha\Big) \\
&= \alpha x^3\Big((c-2)(-1+cx) + c\alpha\Big)
\end{aligned}$$

Note that this determinant can be zero when

$$\alpha = \frac{(c-2)(1-cx)}{c}. \tag{C.4}$$

We show this is not possible by splitting our argument into two parts, one about the convergent regime of the algorithm (where, $\delta\sigma_1^2 < \frac{2(1-\alpha^2)}{c+(c-2)\alpha}$) and the other about the divergent regime.

Let us first provide a proof for the convergent regime of the algorithm. For this regime, let the chosen $\delta$ be represented as $\delta^+$. Now, for the smaller eigen direction, $x = \delta^+\lambda_{\min} = c\delta^+\sigma_1^2/\kappa$. Suppose $\alpha$ was chosen as per equation (C.4),

$$\begin{aligned}
\frac{c\alpha}{c-2} &= 1 - \frac{c^2\delta^+\sigma_1^2}{\kappa} \\
\implies \delta^+\sigma_1^2 &= \frac{\kappa}{c^2} - \frac{\kappa\alpha}{c(c-2)}.
\end{aligned}$$

We will now prove that $\delta^+\sigma_1^2 = \frac{\kappa}{c}(\frac{1}{c} - \frac{\alpha}{c-2})$ is much larger than one allowed by the convergence of the HB updates, i.e., $\delta\sigma_1^2 < \frac{2(1-\alpha^2)}{c+(c-2)\alpha} \leq \frac{2(1-\alpha^2)}{c}$. In particular, if we prove that $\frac{\kappa}{c}(\frac{1}{c} - \frac{\alpha}{c-2}) > \frac{2(1-\alpha^2)}{c}$ for any admissible value of $\alpha$, we are done.

$$\begin{aligned}
\frac{\kappa}{c}(\frac{1}{c} - \frac{\alpha}{c-2}) &> \frac{2(1-\alpha^2)}{c} \\
\Leftrightarrow \frac{\kappa}{c} - \frac{\kappa\alpha}{c-2} &> 2 - 2\alpha^2 \\
\Leftrightarrow \frac{\kappa}{c} - \frac{\kappa\alpha}{c-2} > \frac{\kappa}{c} - \frac{\kappa\alpha}{c} &> 2 - 2\alpha^2 \\
\Leftrightarrow \kappa - \kappa\alpha &> 2c - 2c\alpha^2 \\
\Leftrightarrow 2c\alpha^2 - \kappa\alpha + (\kappa - 2c) &> 0.
\end{aligned}$$

The two roots of this quadratic equation are $\alpha^+ = \frac{\kappa}{2c} - 1$ and $\alpha^- = 1$. Note that $\kappa \geq \widetilde{\kappa} = c$; note that there is not much any method gains over SGD if $\kappa = \mathcal{O}(c)$. And, for any $\kappa \geq 4c$, note, $\alpha^+ > \alpha^-$, indicating that the above equation holds true if $\alpha > \alpha^+ = \frac{\kappa}{2c} - 1$ or if $\alpha < \alpha^- = 1$. The latter condition is true and hence the proposition that $\delta^+\sigma_1^2 > \frac{2(1-\alpha^2)}{c+(c-2)\alpha}$ is true.

We need to prove that the determinant does not vanish in the divergent regime for rounding up the proof to the lemma.

Now, let us consider the divergent regime of the algorithm, i.e., when, $\delta\sigma_1^2 > \frac{2(1-\alpha^2)}{c+(c-2)\alpha}$. Furthermore, for the larger eigendirection, the determinant is zero when $\delta\sigma_1^2 = \frac{1-\frac{c\alpha}{c-2}}{c} = \frac{1}{c} - \frac{\alpha}{c-2}$ (obtained by substituting $x = \delta\sigma_1^2$ in equation (C.4)). If we show that $\frac{2(1-\alpha^2)}{c+(c-2)\alpha} > \frac{1}{c} - \frac{\alpha}{c-2}$ for all admissible values of $c$, we are done. We will explore this in greater detail:

$$\begin{aligned}
\frac{2(1-\alpha^2)}{c+(c-2)\alpha} &> \frac{1}{c} - \frac{\alpha}{c-2} \\
\Leftrightarrow 2(1-\alpha^2) &\geq 1 + \frac{c-2}{c}\alpha - \frac{c}{c-2}\alpha - \alpha^2 \\
\Leftrightarrow 1 - \alpha^2 &\geq \frac{-4(c-1)}{c(c-2)}\alpha \\
\Leftrightarrow c^2 - 2c - \alpha^2c^2 + 2c\alpha^2 &\geq -4c\alpha + 4\alpha \\
\Leftrightarrow c^2(1-\alpha^2) - 2c(1-\alpha^2-2\alpha) - 4\alpha &\geq 0.
\end{aligned}$$

considering the quadratic in the left hand size and solving it for $c$, we have:

$$\begin{aligned}
c^{\pm} &= \frac{2(1-\alpha^2-2\alpha) \pm \sqrt{4(1-\alpha^2-2\alpha)^2+16\alpha(1-\alpha^2)}}{2(1-\alpha^2)} \\
&= \frac{(1-\alpha^2-2\alpha) \pm \sqrt{(1-\alpha^2-2\alpha)^2+4\alpha(1-\alpha^2)}}{(1-\alpha^2)} \\
&= \frac{(1-\alpha^2-2\alpha) \pm \sqrt{1+\alpha^4+4\alpha^2-2\alpha^2-4\alpha+4\alpha^3+4\alpha(1-\alpha^2)}}{(1-\alpha^2)} \\
&= \frac{(1-\alpha^2-2\alpha) \pm (1+\alpha^2)}{(1-\alpha^2)}
\end{aligned}$$

This holds true iff

$$c \leq c^- = \frac{-2\alpha(1+\alpha)}{1-\alpha^2} = \frac{-2\alpha}{1-\alpha},$$

or iff,

$$c \geq c^+ = \frac{2(1-\alpha)}{1-\alpha^2} = \frac{2}{1+\alpha}.$$

Which is true automatically since $c > 2$. This completes the proof of the lemma. □

We are now ready to prove Lemma 46.

*Proof of Lemma 46.* Combining Lemmas 50 and 53, we see that no matter what stepsize and momentum we choose, $\mathcal{B}^{(j)}$ has an eigenvalue of magnitude at least $1 - \frac{500}{\kappa}$ for some $j \in \{1, 2\}$. This proves the lemma. □

## C.2 Equivalence of Algorithm 6 and ASGD

We begin by writing out the updates of ASGD as written out in [54], which starts with two iterates $\widehat{a}_0$ and $\widehat{d}_0$, and from time $t = 0, 1, ...T-1$ implements the following updates:

$$\widehat{b}_t = \alpha_1 \widehat{a}_t + (1-\alpha_1)\widehat{d}_t \tag{C.5}$$

$$\widehat{a}_{t+1} = \widehat{b}_t - \delta_1 \hat{\nabla} f_{t+1}(\widehat{b}_t) \tag{C.6}$$

$$\widehat{c}_t = \beta_1 \widehat{b}_t + (1-\beta_1)\widehat{d}_t \tag{C.7}$$

$$\widehat{d}_{t+1} = \widehat{c}_t - \gamma_1 \hat{\nabla} f_{t+1}(\widehat{b}_t). \tag{C.8}$$

Next, we specify the step sizes $\beta_1 = c_3^2/\sqrt{\kappa\widetilde{\kappa}}$, $\alpha_1 = c_3/(c_3+\beta)$, $\gamma_1 = \beta/(c_3\lambda_{\min})$ and $\delta_1 = 1/R^2$, where $\kappa = R^2/\lambda_{\min}$. Note that the step sizes in the paper of [54] with $c_1$ in their paper set to 1 yields the step sizes above. Now, substituting equation (C.7) in equation (C.8) and substituting the value of $\gamma_1$, we have:

$$\begin{aligned} \widehat{d}_{t+1} &= \beta_1 \left( \widehat{b}_t - \frac{1}{c_3\lambda_{\min}} \hat{\nabla} f_{t+1}(\widehat{b}_t) \right) + (1-\beta_1)\widehat{d}_t \\ &= \beta_1 \left( \widehat{b}_t - \frac{\delta\kappa}{c_3} \hat{\nabla} f_{t+1}(\widehat{b}_t) \right) + (1-\beta_1)\widehat{d}_t. \end{aligned} \tag{C.9}$$

We see that $\widehat{d}_{t+1}$ is precisely the update of the running average $\bar{w}_{t+1}$ in the ASGD method employed in this chapter.

We now update $\widehat{b}_t$ to become $\widehat{b}_{t+1}$ and this can be done by writing out equation (C.5) at $t+1$, i.e:

$$\begin{aligned} \widehat{b}_{t+1} &= \alpha_1 \widehat{a}_{t+1} + (1-\alpha_1)\widehat{d}_{t+1} \\ &= \alpha_1 \left( \widehat{b}_t - \delta_1 \hat{\nabla} f_{t+1}(\widehat{b}_t) \right) + (1-\alpha_1)\widehat{d}_{t+1}. \end{aligned} \tag{C.10}$$

By substituting the value of $\alpha_1$ we note that this is indeed the update of the iterate as a convex combination of the current running average and a short gradient step as written in this chapter. In this chapter, we set $c_3$ to be equal to 0.7, and any constant less than 1 works. In terms of variables, we note that $\alpha$ in this chapter's algorithm description maps to $1-\beta_1$.

## C.3 More details on experiments

In this section, we will present more details on our experimental setup.

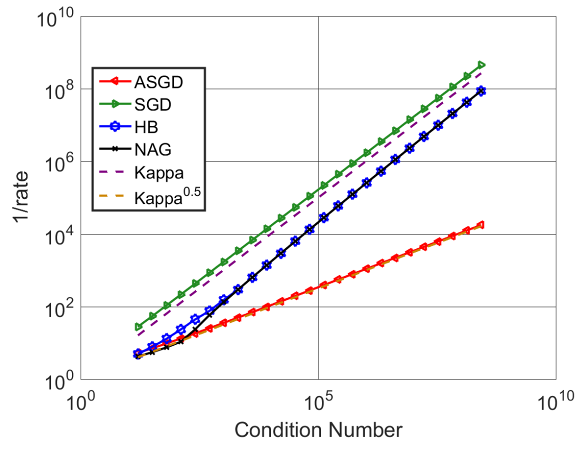
### C.3.1 Linear Regression

In this section, we will present some more results on our experiments on the linear regression problem. Just as in Appendix C.1, it is indeed possible to compute the expected error of all the algorithms among SGD, HB, NAG and ASGD, by tracking certain covariance matrices which evolve as linear systems. For SGD, for instance, denoting

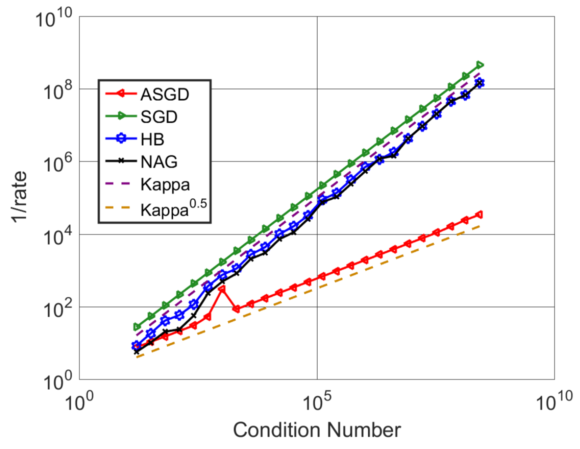
$\mathbf{\Phi}_t^{SGD} \overset{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_t^{SGD} - w^*\right) \otimes \left(\mathbf{w}_t^{SGD} - w^*\right)\right]$, we see that $\mathbf{\Phi}_{t+1}^{SGD} = \mathcal{B} \circ \mathbf{\Phi}_t^{SGD}$, where $\mathcal{B}$ is a linear operator acting on $d \times d$ matrices such that $\mathcal{B} \circ M \overset{\text{def}}{=} M - \delta HM - \delta MH + \delta^2 \mathbb{E}\left[\langle x, Mx \rangle xx^\top\right]$. Similarly, HB, NAG and ASGD also have corresponding operators (see Appendix C.1 for more details on the operator corresponding to HB). The largest magnitude of the eigenvalues of these matrices indicate the rate of decay achieved by the particular algorithm – smaller it is compared to 1, faster the decay.

We now detail the range of parameters explored for these results: the condition number $\kappa$ was varied from $\{2^4, 2^5, .., 2^{28}\}$ for all the optimization methods and for both the discrete and gaussian problem. For each of these experiments, we draw 1000 samples and compute the empirical estimate of the fourth moment tensor. For NAG and HB, we did a very fine grid search by sampling 50 values in the interval $(0, 1]$ for both the learning rate and the momentum parameter and chose the parameter setting that yielded the smallest $\lambda_{\max}(\mathcal{B})$ that is less than 1 (so that it falls in the range of convergence of the algorithm). As for SGD and ASGD, we employed a learning rate of 1/3 for the Gaussian case and a step size of 0.9 for the discrete case. The statistical advantage parameter of ASGD was chosen to be $\sqrt{3\kappa/2}$ for the Gaussian case and $\sqrt{2\kappa/3}$ for the Discrete case, and the a long step parameters of $3\kappa$ and $2\kappa$ were chosen for the Gaussian and Discrete case respectively. The reason it appears as if we choose a parameter above the theoretically maximal allowed value of the advantage parameter is because the definition of $\kappa$ is different in this case. The $\kappa$ we speak about for this experiment is $\lambda_{\max}/\lambda_{\min}$ unlike the condition number for the stochastic optimization problem. In a manner similar to actually running the algorithms (the results of whose are presented in the main chapter), we also note that we can compute the rate as in equation (5.1) and join all these rates using a curve and estimate its slope (in the log scale). This result is indicated in table C.1.

Figure C.1 presents these results, where for each method, we did grid search over all parameters and chose parameters that give smallest $\lambda_{\max}$. Clearly SGD,HB and NAG all have linear dependence on condition number $\kappa$, while ASGD has a dependence of $\sqrt{\kappa}$.



(a) Discrete distribution



(b) Gaussian distribution

Figure C.1: Expected rate of error decay (equation (5.1)) vs condition number for various methods for the linear regression problem. Left is for discrete distribution and right is for Gaussian distribution.

| Algorithm | Slope – discrete | Slope – Gaussian |
|---|---|---|
| SGD | 0.9990 | 0.9995 |
| HB | 1.0340 | 0.9989 |
| NAG | 1.0627 | 1.0416 |
| ASGD | 0.4923 | 0.4906 |

Table C.1: Slopes (i.e. $\gamma$) obtained by fitting a line to the curves in Figure C.1. A value of $\gamma$ indicates that the error decays at a rate of $\exp\left(\frac{-t}{\kappa^{\gamma}}\right)$. A smaller value of $\gamma$ indicates a faster rate of error decay.

### *C.3.2 Autoencoders for MNIST*

We begin by noting that the learning rates tend to vary as we vary batch sizes, which is something that is known in theory [55]. Furthermore, we extend the grid especially whenever our best parameters of a baseline method tends to land at the edge of a grid. The parameter ranges explored by our grid search are:

**Batch Size** 1: (parameters chosen by running for 20 epochs)

- SGD: learning rate: $\{0.01, 0.01\sqrt{10}, 0.1, 0.1\sqrt{10}, 1, \sqrt{10}, 5, 10, 20, 10\sqrt{10}, 40, 60, 80, 100.$
- NAG/HB: learning rate: $\{0.01\sqrt{10}, 0.1, 0.1\sqrt{10}, 1, \sqrt{10}, 10\}$, momentum $\{0, 0.5, 0.75, 0.9, 0.95, 0.97\}$.
- ASGD: learning rate: $\{2.5, 5\}$, long step $\{100.0, 1000.0\}$, advantage parameter $\{2.5, 5.0, 10.0, 20.0\}$.

**Batch Size** 8: (parameters chosen by running for 50 epochs)

- SGD: learning rate: $\{0.001, 0.001\sqrt{10.0}, 0.01, 0.01\sqrt{10}, 0.1, 0.1\sqrt{10}, 1, \sqrt{10}, 5, 10$ $, 10\sqrt{10}, 40, 60, 80, 100, 120, 140\}$.
- NAG/HB: learning rate: $\{5.0, 10.0, 20.0, 10\sqrt{10}, 40, 60\}$, momentum $\{0, 0.25, 0.5, 0.75, 0.9, 0.95\}$.
- ASGD: learning rate $\{40, 60\}$. For a long step of 100, advantage parameters of $\{1.5, 2, 2.5, 5, 10, 20\}$. For a long step of 1000, we swept over advantage parameters of $\{2.5, 5, 10\}$.

### *C.3.3 Deep Residual Networks for CIFAR-10*

In this section, we will provide more details on our experiments on cifar-10, as well as present some additional results. We used a weight decay of 0.0005 in all our experiments. The grid search parameters we used for various algorithms are as follows. Note that the ranges in which parameters such as learning rate need to be searched differ based on batch size [55]. Furthermore, we tend to extrapolate the grid search whenever a parameter (except for the learning rate decay factor) at the edge of the grid has been chosen; this is done so that we always tend to lie in the interior of the grid that we have searched on. Note that for the purposes of the grid search, we choose a hold out set from the training data and add it in to the training data after the parameters are chosen, for the final run.

**Batch Size** 8: Note: (i) parameters chosen by running for 40 epochs and picking the grid search parameter that yields the smallest validation 0/1 error. (ii) The validation set decay scheme that we use is that if the validation error does not decay by at least 1% every three passes over the data, we cut the learning rate by a constant factor (which is grid searched as described below). The minimal learning rate to use is fixed to be $6.25 \times 10^{-5}$, so that we do not decay far too many times and curtail progress prematurely.

- SGD: learning rate: $\{0.0033, 0.01, 0.033, 0.1, 0.33\}$, learning rate decay factor $\{5, 10\}$.
- NAG/HB: learning rate: $\{0.001, 0.0033, 0.01, 0.033\}$, momentum $\{0.8, 0.9, 0.95, 0.97\}$, learning rate decay factor $\{5, 10\}$.
- ASGD: learning rate $\{0.01, 0.0330, 0.1\}$, long step $\{1000, 10000, 50000\}$, advantage parameter $\{5, 10\}$, learning rate decay factor $\{5, 10\}$.

**Batch Size** 128: Note: (i) parameters chosen by running for 120 epochs and picking the grid search parameter that yields the smallest validation 0/1 error. (ii) The validation set decay scheme that we use is that if the validation error does not decay by at least 0.2% every four passes over the data, we cut the learning rate by a constant factor (which is grid searched as described below). The minimal learning rate to use is fixed to be $1 \times 10^{-3}$, so that we do not decay far too many times and curtail progress prematurely.

- SGD: learning rate: $\{0.01, 0.03, 0.09, 0.27, 0.81\}$, learning rate decay factor $\{2, \sqrt{10}, 5\}$.

- NAG/HB: learning rate: $\{0.01, 0.03, 0.09, 0.27\}$, momentum $\{0.5, 0.8, 0.9, 0.95, 0.97\}$, learning rate decay factor $\{2, \sqrt{10}, 5\}$.

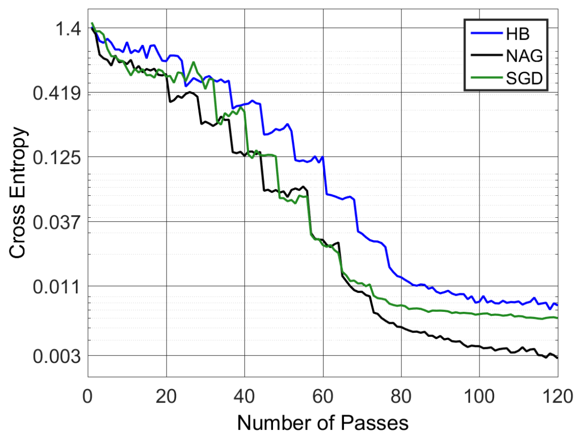
- ASGD: learning rate $\{0.01, 0.03, 0.09, 0.27\}$, long step $\{100, 1000, 10000\}$, advantage parameter $\{5, 10, 20\}$, learning rate decay factor $\{2, \sqrt{10}, 5\}$.

As a final remark, for any comparison across algorithms, such as, (i) ASGD vs. NAG, (ii) ASGD vs HB, we fix the starting learning rate, learning rate decay factor and decay schedule chosen by the best grid search run of NAG/HB respectively and perform a grid search over the long step and advantage parameter of ASGD. In a similar manner, when we compare (iii) SGD vs NAG or, (iv) SGD vs. HB, we choose the learning rate, learning rate decay factor and decay schedule of SGD and simply sweep over the momentum parameter of NAG or HB and choose the momentum that offers the best validation error.

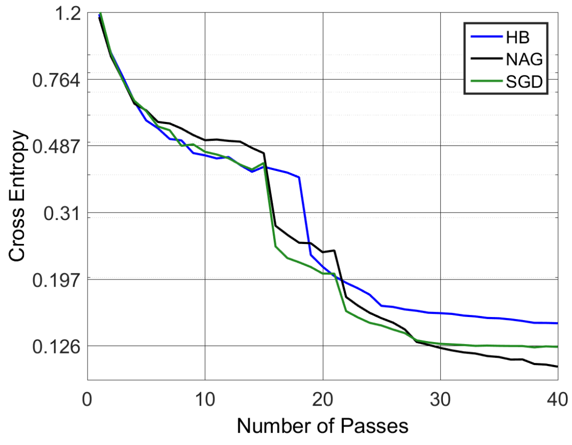
We now present plots of training function value for different algorithms and batch sizes.

**Effect of minibatch sizes**: Figure C.2 plots training function value for batch sizes of 128 and 8 for SGD, HB and NAG. We notice that in the initial stages of training, NAG obtains substantial improvements compared to SGD and HB for batch size 128 but not for batch size 8. Towards the end of training however, NAG starts decreasing the training function value rapidly for both the batch sizes. The reason for this phenomenon is not clear. Note however, that at this point, the test error has already stabilized and the algorithms are just overfitting to the data.

**Comparison of ASGD with momentum methods**: We now present the training error plots for ASGD compared to HB and NAG in Figures C.3 and C.4 respectively. As mentioned earlier, in order to see a clear trend, we constrain the learning rate and decay schedule of ASGD to be the same as that of HB and NAG respectively, which themselves were learned using grid search. We see similar trends as in the validation error plots from Figures 5.5 and 5.6. Please see the figures and their captions for more details.

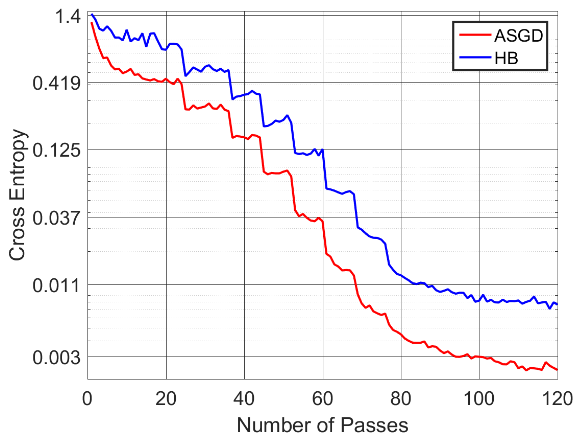


(a) Training Function Value (Batchsize 128)

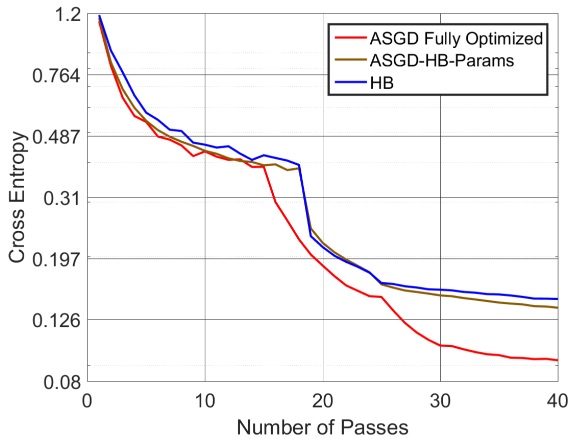


(b) Training Function Value (Batchsize 8)

Figure C.2: Training loss for batch sizes 128 and 8 respectively for SGD, HB and NAG.

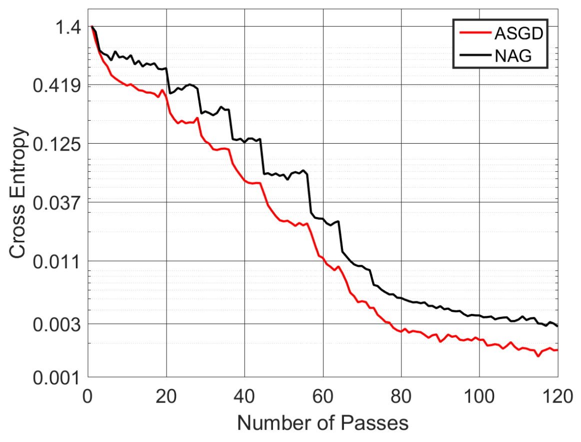


(a) Training Function Value (Batchsize 128)

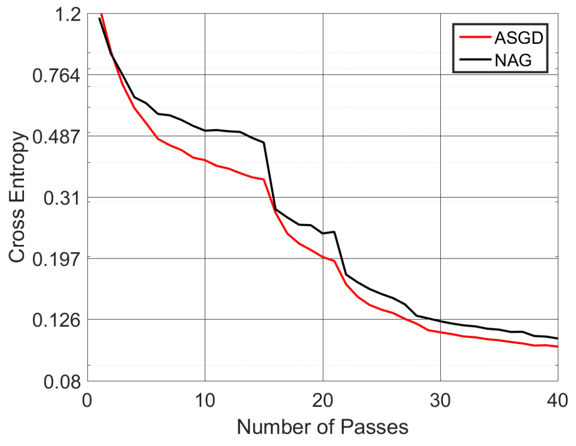


(b) Training Function Value (Batchsize 8)

Figure C.3: Training function value for ASGD compared to HB for batch sizes 128 and 8 respectively.



(a) Training Function Value (Batchsize 128)



(b) Training Function Value (Batchsize 8)

Figure C.4: Training function value for ASGD compared to NAG for batch size 128 and 8 respectively.

# Appendix D

# APPENDIX: SGD'S FINAL ITERATE – MINIMAX OPTIMALITY AND STEPSIZE SCHEMES

## *D.1 Preliminaries*

Before presenting the lemmas establishing the behavior of SGD under various learning rate schemes, we introduce some notation. We recount that the SGD update rule denoted through:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta_t \widehat{\nabla f}(\mathbf{w}_{t-1})$$

We then write out the expression for the stochastic gradient $\widehat{\nabla f}(\mathbf{w}_{t-1})$.

$$\widehat{\nabla f}(\mathbf{w}_{t-1}) = \mathbf{x}_t \mathbf{x}_t^\top (\mathbf{w}_{t-1} - \mathbf{w}^*) - \epsilon_t \mathbf{x}_t,$$

where, given the stochastic gradient corresponding to an example $(\mathbf{x}_t, y_t) \sim \mathcal{D}$, with $y_t = \langle \mathbf{w}^*, \mathbf{x}_t \rangle + \epsilon_t$, the above stochastic gradient expression naturally follows. Now, in order to analyze the contraction properties of the SGD update rule, we require the following notation:

$$P_t = \mathbf{I} - \eta_t \mathbf{x}_t \mathbf{x}_t^\top.$$

**Lemma 55.** *[For e.g. Appendix A.2.2 from [55]]* **Bias-Variance tradeoff:** *Running SGD for $T-$steps starting from $\mathbf{w}_0$ and a stepsize sequence $\{\eta_t\}_{t=1}^T$ presents a final iterate $\mathbf{w}_T$ whose excess risk is upper-bounded as:*

$$\langle \mathbf{H}, \mathbb{E}\left[(\mathbf{w}_T - \mathbf{w}^*) \otimes (\mathbf{w}_T - \mathbf{w}^*)\right] \rangle \leq 2 \cdot \Bigg( \langle \mathbf{H}, \mathbb{E}\left[P_T \cdots P_1 (\mathbf{w}_0 - \mathbf{w}^*) \otimes (\mathbf{w}_0 - \mathbf{w}^*) P_1 \cdots P_T\right] \rangle + \langle \mathbf{H}, \sum_{\tau=1}^{T} \eta_\tau^2 \cdot \mathbb{E}\left[P_T \cdots P_{\tau+1} n_\tau \otimes n_\tau P_{\tau+1} \cdots P_T\right] \rangle \Bigg),$$

*where, $P_t = \mathbf{I} - \eta_t \cdot \mathbf{x}_t \mathbf{x}_t^\top$ and $n_t = \epsilon_t x_t$. Note that $\mathbb{E}\left[n_t | \mathcal{F}_{t-1}\right] = 0$ and $\mathbb{E}\left[n_t \otimes n_t | \mathcal{F}_{t-1}\right] \preceq \sigma^2 \mathbf{H}$, where, $\mathcal{F}_{t-1}$ is the filtration formed by all samples $(\mathbf{x}_1, y_1) \cdots (\mathbf{x}_{t-1}, y_{t-1})$ until time $t$.*

*Proof.* One can view the contribution of the above two terms as ones stemming from SGD's updates, which can be written as:

$$\begin{aligned}
\mathbf{w}_t &= \mathbf{w}_{t-1} - \eta_t \widehat{\nabla f}(w_{t-1}) \\
\mathbf{w}_t - \mathbf{w}^* &= (\mathbf{I} - \eta_t \mathbf{x}_t \mathbf{x}_t)(\mathbf{w}_{t-1} - \mathbf{w}^*) + \eta_t n_t \\
\mathbf{w}_t - \mathbf{w}^* &= P_t \cdots P_1(\mathbf{w}_0 - \mathbf{w}^*) + \sum_{\tau=1}^{T} P_t \cdots P_{\tau+1} \eta_\tau n_\tau
\end{aligned}$$

From the above equation, the result of the lemma follows straightforwardly. Now, in order to bound the total error, note that the original stochastic process associated with SGD's updates can be decoupled into two (simpler) processes, one being the noiseless process (which corresponds to reducing the dependence on the initial error, and is termed "bias"), i.e., where, the recurrence evolves as:

$$\mathbf{w}_t^{\text{bias}} - \mathbf{w}^* = P_t(\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*) \tag{D.1}$$

The second recursion corresponds to the dependence on the noise (termed as variance), wherein, the process is initiated at the solution, i.e. $\mathbf{w}_0^{\text{var}} = \mathbf{w}^*$ and is driven by the noise $n_t$. The update for this process corresponds to:

$$\begin{aligned}
\mathbf{w}_t^{\text{var}} - \mathbf{w}^* &= P_t(\mathbf{w}_{t-1}^{\text{var}} - \mathbf{w}^*) + \eta_t n_t, \quad \text{with} \quad \mathbf{w}_0^{\text{var}} = \mathbf{w}^* \qquad \text{(D.2)} \\
&= \sum_{\tau=1}^{t} P_t \cdots P_{\tau+1} \cdot (\eta_\tau n_\tau).
\end{aligned}$$

We represent by $B_t$ the covariance of the $t^{\text{th}}$ iterate of the bias process, i.e.,

$$\begin{aligned}
B_t &= \mathbb{E}\left[\left(\mathbf{w}_t^{\text{bias}} - \mathbf{w}^*\right)\left(\mathbf{w}_t^{\text{bias}} - \mathbf{w}^*\right)^\top\right] \\
&= \mathbb{E}\left[P_t B_{t-1} P_t^\top\right] = \mathbb{E}\left[P_t \cdots P_1 B_0 P_1 \cdots P_t\right]
\end{aligned}$$

The quantity that routinely shows up when bounding SGD's convergence behavior is the covariance of the variance error, i.e. $V_t := \mathbb{E}\left[(\mathbf{w}_t^{\text{var}} - \mathbf{w}^*) \otimes \mathbf{w}_t^{\text{var}} - \mathbf{w}^*)\right]$. This implies the following (simplified) expression for $V_t$:

$$V_t = \mathbb{E}\left[(\mathbf{w}_t^{\text{var}} - \mathbf{w}^*) \otimes (\mathbf{w}_t^{\text{var}} - \mathbf{w}^*)\right]$$

$$
\begin{aligned}
&= \mathbb{E}\left[\left(\sum_{\tau=1}^{t} P_t \cdots P_{\tau+1} \cdot (\eta_\tau n_\tau)\right) \otimes \left(\sum_{\tau'=1}^{t} P_t \cdots P_{\tau'+1} \cdot (\eta'_\tau n'_\tau)\right)\right] \\
&= \sum_{\tau,\tau'} \mathbb{E}\left[P_T \cdots P_{\tau+1}(\eta_\tau n_\tau) \otimes (\eta_{\tau'} n_{\tau'}) P_{\tau'+1} \cdots P_T\right] \\
&= \sum_{\tau=1}^{T} \eta_\tau^2 \mathbb{E}\left[P_T \cdots P_{\tau+1} n_\tau \otimes n_\tau P_{\tau+1} \cdots P_T\right]
\end{aligned}
$$

Firstly, note that this naturally implies that the sequence of covariances $V_\tau$, $\tau = 1, \cdots, T$ initialized at (say), the solution, i.e., $\mathbf{V}_0 = 0$ naturally grows to its steady state covariance, i.e.,

$$V_1 \preceq V_2 \preceq \cdots \preceq V_\infty.$$

See Lemma 3 of [53] for more details. Furthermore, what naturally follows in relating $V_t$ to $V_{t-1}$ is:

$$V_t \preceq \mathbb{E}\left[P_t V_{t-1} P_t^\top\right] + \eta_t^2 \sigma^2 \mathbf{H}. \tag{D.3}$$

□

**Lemma 56** (Lemma 5 of [53]). *Running SGD with a (constant) stepsize sequence $\eta < 1/R^2$ achieves the following steady-state covariance:*

$$V_\infty \preceq \frac{\eta\sigma^2}{1-\eta R^2}\mathbf{I}.$$

**Lemma 57.** *Suppose $\eta = 1/2R^2$, and $V_0 = \frac{\eta\sigma^2}{1-\eta R^2}\mathbf{I} = 2\eta\sigma^2\mathbf{I}$. For any sequence of learning rates $\eta_t \leq \eta = 1/2R^2 \ \forall\ t \in \{1, \cdots, t\}$, then,*

$$V_t \preceq 2\eta\sigma^2\mathbf{I} \quad \forall \quad t.$$

*Proof.* We will prove the lemma using an inductive argument. The base case, i.e. $t = 0$ follows from the problem statement. Note also that for SGD, $V_0 = 0$ implying the statement naturally follows. If, say, $V_t$ satisfies the equation above, from equation (D.3), we have the following covariance for $V_{t+1}$:

$$V_{t+1} \preceq \mathbb{E}\left[P_t V_t P_t^\top\right] + \eta_t^2 \sigma^2 \mathbf{H}$$

$$\begin{aligned}
&= \mathbb{E}\left[(\mathbf{I} - \eta_t \mathbf{x}_t \mathbf{x}_t^\top) V_t (\mathbf{I} - \eta_t \mathbf{x}_t \mathbf{x}_t^\top)\right] + \eta_t^2 \sigma^2 \mathbf{H} \\
&\preceq 2\eta\sigma^2 \mathbb{E}\left[(\mathbf{I} - \eta_t \mathbf{x}_t \mathbf{x}_t^\top)(\mathbf{I} - \eta_t \mathbf{x}_t \mathbf{x}_t^\top)\right] + \eta_t^2 \sigma^2 \mathbf{H} \\
&\preceq 2\eta\sigma^2 \mathbf{I} - 4\eta_t \eta \sigma^2 \mathbf{H} + 2\eta_t^2 \eta \sigma^2 R^2 \mathbf{H} + \eta_t^2 \sigma^2 \mathbf{H} \\
&\preceq 2\eta\sigma^2 \mathbf{I} - 2\eta_t \eta \sigma^2 \mathbf{H} + \eta_t^2 \sigma^2 \mathbf{H} \\
&= 2\eta\sigma^2 \mathbf{I} + \eta_t \cdot (\eta_t - 2\eta) \sigma^2 \mathbf{H} \\
&\preceq 2\eta\sigma^2 \mathbf{I},
\end{aligned}$$

from which the lemma follows. □

**Lemma 58. (Reduction from Multiplicative noise oracle)** *Let $V_t$ be the (expected) covariance of the variance error. Then, the recursion that connects $V_{t+1}$ to $V_t$ can be expressed as:*

$$V_{t+1} \preceq (\mathbf{I} - \eta_t \mathbf{H}) V_t (\mathbf{I} - \eta_t \mathbf{H}) + 2\eta_t^2 \sigma^2 \mathbf{H}$$

*Proof.* From equation (D.3), we already know that the evolution of the co-variance of the variance error follows:

$$\begin{aligned}
V_{t+1} &\preceq \mathbb{E}\left[P_t V_t P_t^\top\right] + \eta_t^2 \sigma^2 \mathbf{H} \\
&\preceq \mathbb{E}\left[(\mathbf{I} - \eta_t \mathbf{H}) V_t (\mathbf{I} - \eta_t \mathbf{H})\right] + \eta_t^2 \mathbb{E}\left[\mathbf{x}_t \mathbf{x}_t^\top V_t \mathbf{x}_t \mathbf{x}_t^\top\right] + \eta_t^2 \sigma^2 \mathbf{H} \\
&\preceq (\mathbf{I} - \eta_t \mathbf{H}) V_t (\mathbf{I} - \eta_t \mathbf{H}) + \eta_t^2 \left\|V_t\right\|_2 R^2 \mathbf{H} + \eta_t^2 \sigma^2 \mathbf{H} \\
&= (\mathbf{I} - \eta_t \mathbf{H}) V_t (\mathbf{I} - \eta_t \mathbf{H}) + \eta_t^2 \cdot 2\eta\sigma^2 R^2 \mathbf{H} + \eta_t^2 \sigma^2 \mathbf{H} \\
&\preceq (\mathbf{I} - \eta_t \mathbf{H}) V_t (\mathbf{I} - \eta_t \mathbf{H}) + 2\eta_t^2 \sigma^2 \mathbf{H}.
\end{aligned}$$

Where the steps follow from Lemma 57, and owing from the fact that $\eta_t \leq \eta = 1/2R^2 \;\; \forall \;\; t$.
□

**Note:** Basically, one could analyze an auxiliary process driven by noise with variance off by a factor of two and convert the analysis into one involving exact (deterministic) gradients.

**Lemma 59.** **[Bias decay - strongly convex case]** *Let the minimal eigenvalue of the Hessian* $\mu = \lambda_{min}(\mathbf{H}) > 0$. *Consider the bias recursion as in equation* (D.1) *with the stepsize set as* $\eta = 1/(2R^2)$. *Then,*

$$\mathbb{E}\left[\left\|\mathbf{w}_t^{bias} - \mathbf{w}^*\right\|_2^2\right] \leq (1 - 1/(2\kappa))\mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{bias} - \mathbf{w}^*\right\|_2^2\right]$$

*Proof.* The proof follows through straight forward computations:

$$\begin{aligned}\mathbb{E}\left[\left\|\mathbf{w}_t^{\text{bias}} - \mathbf{w}^*\right\|_2^2\right] &\leq \mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*\right\|_2^2\right] - 2\eta\mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*\right\|_{\mathbf{H}}^2\right] + \eta^2 R^2 \mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*\right\|_{\mathbf{H}}^2\right]\\ &= \mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*\right\|_2^2\right] - \eta\mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*\right\|_{\mathbf{H}}^2\right]\\ &\leq (1 - \eta\mu)\mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*\right\|_2^2\right],\end{aligned}$$

where, the first line follows from the fact that $\mathbb{E}\left[\|\mathbf{x}_t\|_2^2 \mathbf{x}_t\mathbf{x}_t^\top\right] \preceq R^2\mathbf{H}$ and the result follows through the definition of $\kappa$. □

**Lemma 60.** **[Reduction of the bias recursion with multiplicative noise to one resembling the variance recursion]** *Consider the bias recursion that evolves as*

$$B_t = \mathbb{E}\left[(\mathbf{w}_t - \mathbf{w}^*)(\mathbf{w}_t - \mathbf{w}^*)^\top\right] = \mathbb{E}\left[(\mathbf{I} - \gamma_t\mathbf{x}_t\mathbf{x}_t^\top)B_{t-1}(\mathbf{I} - \gamma_t\mathbf{x}_t\mathbf{x}_t^\top)\right],$$

*where,* $B_0 = (\mathbf{w}_0 - \mathbf{w}^*)(\mathbf{w}_0 - \mathbf{w}^*)^\top$. *Then, the following recursion holds* $\forall\gamma_t \leq 1/R^2$*:*

$$B_t \preceq (\mathbf{I} - \gamma_t\mathbf{H})B_{t-1}(\mathbf{I} - \gamma_t\mathbf{H}) + \gamma_t^2 R^2 \left\|\mathbf{w}_0 - \mathbf{w}^*\right\|^2 \mathbf{H}.$$

*Proof.* The result follows owing to the following computations:

$$\begin{aligned}B_t &= \mathbb{E}\left[(\mathbf{w}_t - \mathbf{w}^*)(\mathbf{w}_t - \mathbf{w}^*)^\top\right]\\ &= \mathbb{E}\left[(\mathbf{I} - \gamma_t\mathbf{x}_t\mathbf{x}_t^\top)B_{t-1}(\mathbf{I} - \gamma_t\mathbf{x}_t\mathbf{x}_t^\top)\right]\\ &\preceq (\mathbf{I} - \gamma_t\mathbf{H})B_{t-1}(\mathbf{I} - \gamma_t\mathbf{H}) + \gamma_t^2\mathbb{E}\left[(\mathbf{x}_t^\top B_{t-1}\mathbf{x}_t)\mathbf{x}_t\mathbf{x}_t^\top\right]\\ &\preceq (\mathbf{I} - \gamma_t\mathbf{H})B_{t-1}(\mathbf{I} - \gamma_t\mathbf{H}) + \gamma_t^2\mathbb{E}\left[\|B_{t-1}\|_2\right]R^2\mathbf{H}\\ &\preceq (\mathbf{I} - \gamma_t\mathbf{H})B_{t-1}(\mathbf{I} - \gamma_t\mathbf{H}) + \gamma_t^2\mathbb{E}\left[\|\mathbf{w}_{t-1} - \mathbf{w}^*\|_2^2\right]R^2\mathbf{H}\\ &\preceq (\mathbf{I} - \gamma_t\mathbf{H})B_{t-1}(\mathbf{I} - \gamma_t\mathbf{H}) + \gamma_t^2\mathbb{E}\left[\|\mathbf{w}_0 - \mathbf{w}^*\|_2^2\right]R^2\mathbf{H},\end{aligned}$$

with the last inequality holding true if the squared distance to the optimum doesn't grow as a part of the recursion. We prove that this indeed is the case below:

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{w}_{t-1}-\mathbf{w}^*\|_2^2\right] &= \mathbb{E}\left[\left\|\mathbf{w}_{t-2}-\gamma_{t-1}\mathbf{x}_{t-1}\mathbf{x}_{t-1}^\top-\mathbf{w}^*\right\|_2^2\right]\\ &\leq \mathbb{E}\left[\|\mathbf{w}_{t-2}-\mathbf{w}^*\|_2^2\right]-2\gamma_{t-1}\mathbb{E}\left[\|\mathbf{w}_{t-2}-\mathbf{w}^*\|_{\mathbf{H}}^2\right]+\gamma_{t-1}^2R^2\mathbb{E}\left[\|\mathbf{w}_{t-2}-\mathbf{w}^*\|_{\mathbf{H}}^2\right]\\ &\leq \mathbb{E}\left[\|\mathbf{w}_{t-2}-\mathbf{w}^*\|_2^2\right]-\gamma_{t-1}\mathbb{E}\left[\|\mathbf{w}_{t-2}-\mathbf{w}^*\|_{\mathbf{H}}^2\right]\\ &\leq \mathbb{E}\left[\|\mathbf{w}_{t-2}-\mathbf{w}^*\|_2^2\right].\end{aligned}$$

Recursively applying the above argument yields the desired result. □

**Note:** This result implies that the bias error (in the smooth non-strongly convex case of the least squares regression with multiplicative noise) can be bounded by employing a similar lemma as that of the variance, where one can look at the quantity $R^2 \cdot \|\mathbf{w}_0-\mathbf{w}^*\|_2^2$ as the analog of the variance $\sigma^2$ that drives the process.

**Lemma 61.** *[***Reduction of multiplicative noise to additive noise oracle for lower-bounds***] Under the assumption that the covariance of noise $\Sigma=\sigma^2\mathbf{H}$, the following statement holds. Let $V_t$ be the (expected) covariance of the variance error. Then, the recursion that connects $V_{t+1}$ to $V_t$ can be expressed as:*

$$V_{t+1}=\mathbb{E}\left[(\mathbf{I}-\eta_t\mathbf{x}_t\mathbf{x}_t^\top)V_t(\mathbf{I}-\eta_t\mathbf{x}_t\mathbf{x}_t^\top)\right]+\eta_t^2\sigma^2\mathbf{H}$$

*Then,*

$$V_{t+1}\succeq(\mathbf{I}-\eta_t\mathbf{H})V_t(\mathbf{I}-\eta_t\mathbf{H})+\eta_t^2\sigma^2\mathbf{H}$$

*Proof.* Let us consider firstly, the setting of (bounded) additive noise. Here, we have:

$$\hat{\nabla}f(\mathbf{w}_t)=\mathbf{H}(\mathbf{w}_t-\mathbf{w}^*)+\zeta_t,\ \text{with } \mathbb{E}\left[\zeta_t|\mathbf{w}_t\right]=0, \text{ and } \mathbb{E}\left[\zeta_t\zeta_t^\top|\mathbf{w}_t\right]=\sigma^2\mathbf{H}.$$

Then, updates leading upto time $t+1$ can be written as:

$$\mathbf{w}_{t+1}-\mathbf{w}^*=\prod_{\tau=1}^{t+1}(\mathbf{I}-\eta_\tau\mathbf{H})(\mathbf{w}_0-\mathbf{w}^*)+\sum_{\tau'=1}^{t+1}\eta_{\tau'}\prod_{\tau=\tau'+1}^{t+1}(\mathbf{I}-\eta_\tau\mathbf{H})\zeta_{\tau'}$$

This implies the covariance of the variance error is:

$$\begin{aligned}
\tilde{V}_{t+1} &= \mathbb{E}\left[\left(\sum_{\tau'=1}^{t+1}\eta_{\tau'}\prod_{\tau=\tau'+1}^{t+1}(\mathbf{I}-\eta_\tau\mathbf{H})\zeta_{\tau'}\right)\otimes\left(\sum_{\tau''=1}^{t+1}\eta_{\tau''}\prod_{\tau=\tau''+1}^{t+1}(\mathbf{I}-\eta_\tau\mathbf{H})\zeta_{\tau''}\right)\right]\\
&= \sum_{\tau'=1}^{t+1}\eta_{\tau'}^2\mathbb{E}\left[\prod_{\tau=\tau'+1}^{t+1}(\mathbf{I}-\eta_\tau\mathbf{H})\zeta_{\tau'}\otimes\zeta_{\tau'}\prod_{\tau=t+1}^{\tau'+1}(\mathbf{I}-\eta_\tau\mathbf{H})\right]\\
&= (\mathbf{I}-\eta_{t+1}\mathbf{H})V_t(\mathbf{I}-\eta_{t+1}\mathbf{H})+\eta_{t+1}^2\sigma^2\mathbf{H}.
\end{aligned}$$

Now, let us consider the statement of the lemma:

$$\begin{aligned}
V_{t+1} &= \mathbb{E}\left[(\mathbf{I}-\eta_{t+1}\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top)V_t(\mathbf{I}-\eta_{t+1}\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top)\right]+\eta_{t+1}^2\sigma^2\mathbf{H}\\
&= (\mathbf{I}-\eta_{t+1}\mathbf{H})V_t(\mathbf{I}-\eta_{t+1}\mathbf{H})+\eta_{t+1}^2\mathbb{E}\left[(\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top-\mathbf{H})V_t(\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top-\mathbf{H})\right]+\eta_{t+1}^2\sigma^2\mathbf{H}\\
&\succeq (\mathbf{I}-\eta_{t+1}\mathbf{H})V_t(\mathbf{I}-\eta_{t+1}\mathbf{H})+\eta_t^2\sigma^2\mathbf{H}.
\end{aligned}$$

Unrolling the above argument, we see that $V_{t+1}\succeq\tilde{V}_{t+1}$, implying that the process driven by the multiplicative noise oracle can be lower bounded (in a PSD sense) by one that employs deterministic gradients with additive noise. □

## D.2 Proofs of results in Section 6.4.1

**Theorem 62.** *Consider the additive noise oracle setting, where, we have access to stochastic gradients satisfying:*

$$\widehat{\nabla f}(\mathbf{w})=\nabla f(\mathbf{w})+\zeta=\mathbf{H}(\mathbf{w}-\mathbf{w}^*)+\zeta,$$

*where,*

$$\mathbb{E}\left[\zeta|\mathbf{w}\right]=0,\ \textit{and,}\ \mathbb{E}\left[\zeta\zeta^\top|\mathbf{w}\right]=\sigma^2\mathbf{H}$$

*The following lower bounds hold on the final iterate of a Stochastic Gradient procedure with access to the above stochastic gradients when using polynomially decaying stepsizes.*
***Strongly convex case**: Suppose $\mu>0$. For any condition number $\kappa$, there exists a problem instance with initial suboptimality $f(\mathbf{w}_0)-f(\mathbf{w}^*)\leq\sigma^2 d$ such that, for any $T\geq\kappa^{\frac{4}{3}}$, and for*

*all $a, b \geq 0$ and $0.5 \leq \alpha \leq 1$, and for the learning rate scheme $\eta_t = \frac{a}{b+t^\alpha}$, we have*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \geq \exp\left(-\frac{T}{\kappa \log T}\right)\left(f(\mathbf{w}_0) - f(\mathbf{w}^*)\right) + \frac{\sigma^2 d}{64} \cdot \frac{\kappa}{T}.$$

***Smooth case**: For any fixed $T > 1$, there exists a problem instance such that, for all $a, b \geq 0$ and $0.5 \leq \alpha \leq 1$, and for the learning rate scheme $\eta_t = \frac{a}{b+t^\alpha}$, we have*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \geq \left(L \cdot \|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sigma^2 d\right) \cdot \frac{1}{\sqrt{T}\log T}.$$

*Proof.* **Strongly convex case**: The problem instance is simple. Consider the case where the inputs are such that in every example $\mathbf{x}$, there is only one co-ordinate that is non-zero. Furthermore, let each co-ordinate be Gaussian with mean zero and variance for the first $d/2$ co-ordinates be $d\kappa/3$ whereas the rest be 1. This implies $\mathbf{H} = \begin{bmatrix} d\kappa/3 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \ddots \end{bmatrix}$, where the first $\frac{d}{2}$ diagonal entries are equal to $\kappa/3$ and the remaining $\frac{d}{2}$ diagonal entries are equal to 1 and all the off diagonal entries are equal to zero. Furthermore, consider the noise to be additive (and independent of $\mathbf{x}$) with mean zero. Finally, let us denote by $v_t^{(i)} \overset{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_t^{(i)} - (\mathbf{w}^*)^{(i)}\right)^2\right]$ the variance in the $i^{\text{th}}$ direction at time step $t$. Let the initialization be such that $v_0^{(i)} = 3\sigma^2/\kappa$ for $i = 1, 2, ..., d/2$ and $v_0^{(i)} = \sigma^2$ for $i = d/2+1, ..., d$. This means that the variances for all directions with eigenvalue $\kappa$ remain equal as $t$ progresses and similarly for all directions with eigenvalue 1. We have

$$v_T^{(1)} \overset{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(1)} - (\mathbf{w}^*)^{(1)}\right)^2\right] = \prod_{j=1}^{T}\left(1 - \eta_j\kappa/3\right)^2 v_0^{(1)} + \kappa\sigma^2/3\sum_{j=1}^{T}\eta_j^2\prod_{i=j+1}^{T}\left(1 - \eta_i\kappa/3\right)^2 \text{ and}$$

$$v_T^{(d)} \overset{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(d)} - (\mathbf{w}^*)^{(d)}\right)^2\right] = \prod_{j=1}^{T}\left(1 - \eta_j\right)^2 v_0^{(d)} + \sigma^2\sum_{j=1}^{T}\eta_j^2\prod_{i=j+1}^{T}\left(1 - \eta_i\right)^2.$$

We consider a recursion for $v_t^{(i)}$ with eigenvalue $\lambda_i$ ($\kappa$ or 1). By the design of the algorithm, we know

$$v_{t+1}^{(i)} = (1 - \eta_t\lambda_i)^2 v_t^{(i)} + \lambda_i\sigma^2\eta_t^2.$$

Let $s(\eta,\lambda) = \frac{\lambda\sigma^2\eta^2}{1-(1-\eta\lambda)^2}$ be the solution to the stationary point equation $x = (1-\eta\lambda)^2 + \lambda\sigma^2\eta^2$. Intuitively if we keep using the same learning rate $\eta$, then $v_t^{(i)}$ is going to converge to $s(\eta,\lambda_i)$. Also note that $s(\eta,\lambda) \approx \sigma^2\eta/2$ when $\eta\lambda \ll 1$.

We first prove the following claim showing that eventually the variance in direction $i$ is going to be at least $s(\eta_T,\lambda_i)$.

**Remark 1.** *Suppose $s(\eta_t,\lambda_i) \le v_0^{(i)}$, then $v_t^{(i)} \ge s(\eta_t,\lambda_i)$.*

*Proof.* We can rewrite the recursion as

$$v_{t+1}^{(i)} - s(\eta_t,\lambda_i) = (1-\eta_t\lambda_i)^2(v_t^{(i)} - s(\eta_t,\lambda_i)).$$

In this form, it is easy to see that the iteration is a contraction towards $s(\eta_t,\lambda_i)$. Further, $v_{t+1}^{(i)} - s(\eta_t,\lambda_i)$ and $v_t^{(i)} - s(\eta_t,\lambda_i)$ have the same sign. In particular, let $t_0$ be the first time such that $s(\eta_t,\lambda_i) \le v_0^{(i)}$ (note that $\eta_t$ is monotone and so is $s(\eta_t,\lambda_i)$), it is easy to see that $v_t^{(i)} \ge v_0^{(i)}$ when $t \le t_0$. Therefore we know $v_{t_0}^{(i)} \ge s(\eta_{t_0},\lambda_i)$, by the recursion this implies $v_{t_0+1}^{(i)} \ge s(\eta_{t_0},\lambda_i) \ge s(\eta_{t_0+1},\lambda_i)$. The claim then follows from a simple induction. □

If $s(\eta_T,\lambda_i) \ge v_0^{(i)}$ for $i=1$ or $i=d$ then the error is at least $\sigma^2 d/2 \ge \kappa\sigma^2 d/T$ and we are done. Therefore we must have $s(\eta_T,\kappa) \le v_0^{(1)} = 3\sigma^2/\kappa$, and by Claim 1 we know $v_T^{(1)} \ge s(\eta_T,\kappa) \ge \sigma^2\eta_T/2$. The function value is at least

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \ge \frac{d}{2}\cdot\kappa\cdot v_T^{(1)} \ge \frac{d\kappa\sigma^2\eta_T}{12}.$$

To make sure $\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \le \frac{d\kappa\sigma^2}{64T}$ we must have $\eta_T \le \frac{1}{6T}$. Next we will show that when this happens, $v_T^{(d)}$ must be large so the function value is still large.

We will consider two cases, in the first case, $b \ge T^\alpha$. Since $\frac{1}{16T} \ge \eta_T = \frac{a}{b+T^\alpha} \ge \frac{a}{2b}$, we have $\frac{a}{b} \le \frac{1}{8T}$. Therefore $v_T^{(d)} \ge (1-\frac{a}{b})^{2T}v_0^{(d)} \ge \sigma^2/2$, so the function value is at least $\mathbb{E}\left[f(\mathbf{w}_t)\right] \ge \frac{d}{2}\cdot v_T^{(d)} \ge \frac{d\sigma^2}{4} \ge \frac{\kappa d\sigma^2}{T}$, and we are done.

In the second case, $b < T^\alpha$. Since $\frac{1}{16T} \ge \eta_T = \frac{a}{b+T^\alpha} \ge \frac{a}{2T^\alpha}$, we have $a \le \frac{1}{8}T^{\alpha-1}$. The sum of learning rates satisfy

$$\sum_{i=1}^{T} \eta_i \leq \sum_{i=1}^{T} \frac{a}{i^\alpha} \leq \sum_{i=1}^{T} \frac{1}{8} i^{-1} \leq 0.125 \log T.$$

Here the second inequality uses the fact that $T^{\alpha-1} i^{-\alpha} \leq i^{-1}$ when $i \leq T$. Similarly, we also know $\sum_{i=1}^{T} \eta_i^2 \leq \sum_{i=1}^{T} (0.125)^2 i^{-2} \leq \pi^2/384$. Using the approximation $(1-u)^2 \geq \exp(-2u - 4u^2)$ for $u < 1/4$, we get $v_T^{(d)} \geq \exp(-2\sum_{i=1}^{T} \eta_i - 4\sum_{i=1}^{T} \eta_i^2) v_0^{(d)} \geq \sigma^2/5T^{\frac{1}{4}}$, so the function value is at least $\mathbb{E}\left[f(\mathbf{w}_t)\right] \geq \frac{d}{2} \cdot v_T^{(d)} \geq \frac{d\sigma^2}{10T^{\frac{1}{4}}} \geq \frac{\kappa d \sigma^2}{32T}$. This concludes the second case and proves the strongly convex part of the theorem.

**Smooth case**: The proof of this part is quite similar to that of the strongly convex case above but with a subtle change in the initialization. In order to make this clear, we will do the proof from scratch with out borrowing anything from the previous argument. Let $\mathbf{H} = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & \frac{d}{\kappa} & \\ & & & \ddots \end{bmatrix}$, where the first $\frac{d}{2}$ diagonal entries are equal to 1 and the remaining $\frac{d}{2}$ diagonal entries are equal to $\frac{d}{\kappa}$ and all the off diagonal entries are equal to zero. We will use $\kappa = \frac{1}{\sqrt{T}}$. Let us denote by $v_t^{(i)} \overset{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_t^{(i)} - (\mathbf{w}^*)^{(i)}\right)^2\right]$ the variance in the $i^{\text{th}}$ direction at time step $t$. Let the initialization be such that $v_0^{(i)} = \sigma^2/\kappa$ for $i = 1, 2, ..., d/2$ and $v_0^{(i)} = \sigma^2$ for $i = d/2+1, ..., d$. This means that the variances for all directions with eigenvalue $\kappa$ remain equal as $t$ progresses and similarly for all directions with eigenvalue 1. We have

$$v_T^{(1)} \overset{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(1)} - (\mathbf{w}^*)^{(1)}\right)^2\right] = \prod_{j=1}^{T} (1 - \eta_j \kappa/3)^2 \, v_0^{(1)} + \kappa\sigma^2/3 \sum_{j=1}^{T} \eta_j^2 \prod_{i=j+1}^{T} (1 - \eta_i \kappa/3)^2 \text{ and}$$

$$v_T^{(d)} \overset{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(d)} - (\mathbf{w}^*)^{(d)}\right)^2\right] = \prod_{j=1}^{T} (1 - \eta_j)^2 \, v_0^{(d)} + \sigma^2 \sum_{j=1}^{T} \eta_j^2 \prod_{i=j+1}^{T} (1 - \eta_i)^2 .$$

We consider a recursion for $v_t^{(i)}$ with eigenvalue $\lambda_i$ (1 or $\frac{1}{\kappa}$). By the design of the algorithm, we know

$$v_{t+1}^{(i)} = (1 - \eta_t \lambda_i)^2 v_t^{(i)} + \lambda_i \sigma^2 \eta_t^2.$$

Let $s(\eta, \lambda) = \frac{\lambda\sigma^2\eta^2}{1-(1-\eta\lambda)^2}$ be the solution to the stationary point equation $x = (1-\eta\lambda)^2 + \lambda\sigma^2\eta^2$.

Intuitively if we keep using the same learning rate $\eta$, then $v_t^{(i)}$ is going to converge to $s(\eta, \lambda_i)$. Also note that $s(\eta, \lambda) \approx \sigma^2\eta/2$ when $\eta\lambda \ll 1$.

If $s(\eta_T, \lambda_i) \geq v_0^{(i)}$ for $i = 1$ or $i = d$ then the error is at least $\sigma^2 d/2\kappa \geq \kappa\sigma^2 d/T$ and we are done. Therefore we must have $s(\eta_T, \kappa) \leq v_0^{(1)} = 3\sigma^2/\kappa$, and by Claim 1 we know $v_T^{(1)} \geq s(\eta_T, \kappa) \geq \sigma^2\eta_T/2$. The function value is at least

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \geq \frac{d}{2} \cdot v_T^{(1)} \geq \frac{d\sigma^2\eta_T}{4}.$$

To make sure $\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \leq \frac{d\kappa\sigma^2}{64T\log T}$ we must have $\eta_T \leq \frac{\kappa}{16T\log T}$. Next we will show that when this happens, $v_T^{(d)}$ must be large so the function value is still large.

We will consider two cases, in the first case, $b \geq T^\alpha$. Since $\frac{\kappa}{16T\log T} \geq \eta_T = \frac{a}{b+T^\alpha} \geq \frac{a}{2b}$, we have $\frac{a}{b} \leq \frac{\kappa}{8T\log T}$. Therefore $v_T^{(d)} \geq (1 - \frac{a}{b})^{2T} v_0^{(d)} \geq \sigma^2/2$, so the function value is at least $\mathbb{E}\left[f(\mathbf{w}_t)\right] - f(\mathbf{w}^*) \geq \frac{d}{2} \cdot \frac{1}{\kappa} \cdot v_T^{(d)} \geq \frac{d\sigma^2}{4\kappa} \geq \frac{\kappa d\sigma^2}{T}$, and we are done.

In the second case, $b < T^\alpha$. Since $\frac{\kappa}{16T\log T} \geq \eta_T = \frac{a}{b+T^\alpha} \geq \frac{a}{2T^\alpha}$, we have $a \leq \frac{1}{8\log T}\kappa T^{\alpha-1}$. The sum of learning rates satisfy

$$\sum_{i=1}^{T} \eta_i \leq \sum_{i=1}^{T} \frac{a}{i^\alpha} \leq \sum_{i=1}^{T} \frac{1}{8\log T}\kappa i^{-1} \leq 0.125\kappa.$$

Here the second inequality uses the fact that $T^{\alpha-1}i^{-\alpha} \leq i^{-1}$. Similarly, we also know

$$\sum_{i=1}^{T} \eta_i^2 \leq \sum_{i=1}^{T} (0.125\kappa/\log T)^2 i^{-2} \leq \pi^2\kappa^2/384.$$

Using the approximation $(1-u)^2 \geq \exp(-2u - 4u^2)$ for $u < 1/4$, we get $v_T^{(d)} \geq \exp(-2\sum_{i=1}^T \frac{\eta_i}{\kappa} - 4\sum_{i=1}^T \frac{\eta_i^2}{\kappa^2}) v_0^{(d)} \geq \sigma^2/5$, so the function value is at least $\mathbb{E}\left[f(\mathbf{w}_t)\right] \geq \frac{d}{2} \cdot \frac{1}{\kappa} \cdot v_T^{(d)} \geq \frac{d\sigma^2}{10\kappa} \geq \frac{d\sigma^2}{10\sqrt{T}}$. This concludes the second case and proves the strongly convex part of the theorem. Since $\|\mathbf{H}\| \cdot \|\mathbf{w}_0 - \mathbf{w}^*\|^2 = d\sigma^2$, we have

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \geq \sigma^2 d \cdot \min\left(\frac{\kappa}{T\log T}, \frac{1}{10\sqrt{T}}\right) \geq \left(L \cdot \|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sigma^2 d\right) \cdot \frac{1}{\sqrt{T}\log T}.$$

This proves the theorem. □

*Proof of Theorem 18.* The proof of Theorem 18 follows straightforwardly when combining the result of Lemma 61 and Theorem 62. □

## D.3 Proofs of results in Section 6.4.2

**Theorem 63.** *Consider the additive noise oracle setting, where, we have access to stochastic gradients satisfying:*

$$\widehat{\nabla f}(\mathbf{w}) = \nabla f(\mathbf{w}) + \zeta = \mathbf{H}(\mathbf{w} - \mathbf{w}^*) + \zeta,$$

*where,*

$$\mathbb{E}\left[\zeta|\mathbf{w}\right] = 0, \text{ and, } \mathbb{E}\left[\zeta\zeta^\top|\mathbf{w}\right] \preceq \hat{\sigma}^2\mathbf{H}$$

*Running Algorithm 7 with an initial stepsize of $\eta_1 = 1/R^2$, starting from the solution, i.e. $\mathbf{w}_0 = \mathbf{w}^*$ allows the algorithm to obtain the following dependence on the variance error:*

$$\mathbb{E}\left[f(\mathbf{w}_T^{var})\right] - f(\mathbf{w}^*) \leq 2\frac{d\hat{\sigma}^2 \log T}{T}$$

*Proof.* The learning rate scheme is as follows. Divide the total time horizon $T$ into $\log T$ phases, each of length $\frac{T}{\log T}$. In the $\ell$th phase, the learning rate is set to be $\frac{1}{2^\ell R^2}$. The variance in the $k$th coordinate can be bounded as

$$\begin{aligned}
v_T^{(k)} &\leq \prod_{j=1}^{T}\left(1-\eta_j\lambda^{(k)}\right)^2 v_0^{(k)} + \lambda^{(k)}\hat{\sigma}^2\sum_{j=1}^{T}\eta_j^2\prod_{i=j+1}^{T}\left(1-\eta_i\lambda^{(k)}\right)^2 \\
&\leq \exp\left(-2\sum_{j=1}^{T}\eta_j\lambda^{(k)}\right)v_0^{(k)} \\
&\quad + \lambda^{(k)}\hat{\sigma}^2\sum_{\ell=1}^{\log T}\frac{1}{2^{2\ell}(R^2)^2}\sum_{j=1}^{T/\log T}\left(1-\frac{\lambda^{(k)}}{2^\ell(R^2)}\right)^{2j}\cdot\prod_{u=\ell+1}^{\log T}\left(1-\frac{\lambda^{(k)}}{2^u R^2}\right)^{T/\log T} \\
&\leq \exp\left(-\frac{2\lambda^{(k)}}{R^2}\cdot\frac{T}{\log T}\right)v_0^{(k)} + \lambda^{(k)}\hat{\sigma}^2\sum_{\ell=1}^{\log T}\frac{1}{2^{2\ell}(R^2)^2}\cdot\frac{2^\ell R^2}{\lambda^{(k)}}\cdot\prod_{u=\ell+1}^{\log T}\exp\left(-\frac{\lambda^{(k)}T}{2^u R^2\log T}\right) \\
&\leq \exp\left(-\frac{2\lambda^{(k)}}{R^2}\cdot\frac{T}{\log T}\right)v_0^{(k)} + \sum_{\ell=1}^{\log T}\frac{\hat{\sigma}^2}{2^\ell R^2}\prod_{u=\ell+1}^{\log T}\exp\left(-\frac{\lambda^{(k)}T}{2^u R^2\log T}\right). \qquad \text{(D.4)}
\end{aligned}$$

Let $\ell^* \overset{\text{def}}{=} \max\left(0, \lfloor\log\left(\frac{\lambda^{(k)}}{R^2}\cdot\frac{T}{\log T}\right)\rfloor\right)$. We now split the summation in the second term in (D.4) into two parts and bound each of them below.

$$\sum_{\ell=1}^{\ell^*}\frac{\hat{\sigma}^2}{2^\ell R^2}\prod_{u=\ell+1}^{\log T}\exp\left(-\frac{\lambda^{(k)}T}{2^u R^2\log T}\right) \leq \sum_{\ell=1}^{\ell^*}\frac{\hat{\sigma}^2}{2^\ell R^2}\prod_{u=\ell+1}^{\ell^*}\exp\left(-\frac{\lambda^{(k)}T}{2^u R^2\log T}\right)$$

$$\leq \sum_{\ell=1}^{\ell^*} \frac{\hat{\sigma}^2}{2^\ell R^2} \prod_{u=\ell+1}^{\ell^*} \exp\left(-2^{\ell^*-u}\right) \leq \sum_{\ell=1}^{\ell^*} \frac{\hat{\sigma}^2}{2^\ell R^2} \exp\left(-2^{\ell^*-\ell}\right)$$
$$\leq \frac{\hat{\sigma}^2}{2^{\ell^*} R^2} \sum_{\ell=1}^{\ell^*} 2^{\ell^*-\ell} \exp\left(-2^{\ell^*-\ell}\right) \leq \frac{\hat{\sigma}^2}{2^{\ell^*} R^2} \leq \frac{\hat{\sigma}^2}{\lambda^{(k)}} \cdot \frac{\log T}{T}. \tag{D.5}$$

For the second part, we have

$$\sum_{\ell=\ell^*+1}^{\log T} \frac{\hat{\sigma}^2}{2^\ell R^2} \prod_{u=\ell+1}^{\log T} \exp\left(-\frac{\lambda^{(k)} T}{2^u R^2 \log T}\right) \leq \sum_{\ell=\ell^*+1}^{\log T} \frac{\hat{\sigma}^2}{2^\ell R^2} \leq \sum_{\ell=\ell^*+1}^{\log T} \frac{\hat{\sigma}^2}{2^{\ell^*} R^2} \leq \frac{\hat{\sigma}^2}{\lambda^{(k)}} \cdot \frac{\log T}{T}. \tag{D.6}$$

Plugging (D.5) and (D.6) into (D.4), we obtain

$$v_T^{(k)} \leq \exp\left(-\frac{2\lambda^{(k)}}{R^2} \cdot \frac{T}{\log T}\right) v_0^{(k)} + \frac{2\hat{\sigma}^2}{\lambda^{(k)}} \cdot \frac{\log T}{T}.$$

The function suboptimality can now be bounded as

$$\mathbb{E}\left[f(\mathbf{w}_T^{\text{var}})\right] - f(\mathbf{w}^*) = \sum_{k=1}^{d} \lambda^{(k)} \cdot v_T^{(k)}$$
$$\leq \sum_{k=1}^{d} \lambda^{(k)} \left(\exp\left(-\frac{2\lambda^{(k)}}{R^2} \cdot \frac{T}{\log T}\right) v_0^{(k)} + \frac{2\hat{\sigma}^2}{\lambda^{(k)}} \cdot \frac{\log T}{T}\right).$$

$$\mathbb{E}\left[f(\mathbf{w}_T^{\text{var}})\right] - f(\mathbf{w}^*) \leq \sum_{k=1}^{d} \left(\frac{L \log T}{T} v_0^{(k)} + 2\hat{\sigma}^2 \cdot \frac{\log T}{T}\right) = 2\left(\hat{\sigma}^2 d\right) \frac{\log T}{T}.$$

□

*Proof of Theorem 19.* **Smooth case:** The result follows by instantiating $\hat{\sigma}^2$ in Theorem 63 with $2\sigma^2$ (Lemma 58) and $R^2 \|\mathbf{w}_0 - \mathbf{w}^*\|_2^2$ (Lemma 60) and appealing to Lemma 55.

**Strongly convex case:** As with the smooth case, the result relies on instantiating $\hat{\sigma}^2$ in Theorem 63 with $2\sigma^2$ (Lemma 58) and using Lemma 59 and then appealing to Lemma 55. □

**Proposition 64.** *Consider the additive noise oracle setting, where, we have access to stochastic gradients satisfying:*

$$\widehat{\nabla f}(\mathbf{w}) = \nabla f(\mathbf{w}) + \zeta = \mathbf{H}(\mathbf{w} - \mathbf{w}^*) + \zeta,$$

*where,*

$$\mathbb{E}\left[\zeta|\mathbf{w}\right] = 0, \textit{ and, } \mathbb{E}\left[\zeta\zeta^\top|\mathbf{w}\right] \preceq \sigma^2\mathbf{H}$$

*There exists a stepsize scheme with which, by starting at the solution (i.e.* $\mathbf{w}_0 = \mathbf{w}^*$*) the algorithm obtains the following dependence on the variance error, under the assumption that* $\mu > 0$ *and* $\kappa \geq 2$*.*

$$\mathbb{E}\left[f(\mathbf{w}_T^{var})\right] - f(\mathbf{w}^*) \leq 50\log_2\kappa \cdot \frac{\sigma^2 d}{T}.$$

*Proof.* The learning rate scheme is as follows.

We first break $T$ into three equal sized parts. Let $A = T/3$ and $B = 2T/3$. In the first $T/3$ steps, we use a constant learning rate of $1/R^2$. Note that at the end of this phase, (since $T > \kappa$) the dependence on the initial error decays geometrically. In the second $T/3$ steps, we use a polynomial decay learning rate $\eta_{A+t} = \frac{1}{\mu(\kappa+t/2)}$. In the third $T/3$ steps, we break the steps into $\log_2(\kappa)$ equal sized phases. In the $\ell^{\text{th}}$ phase, the learning rate to be used is $\frac{5\log_2\kappa}{2^\ell\cdot\mu\cdot T}$. Note that the learning rate in the first phase depends on strong convexity and that in the last phase depends on smoothness (since the last phase has $\ell = \log\kappa$).

Recall the variance in the $k^{\text{th}}$ coordinate can be upper bounded by

$$\begin{aligned}
v_T^{(k)} \overset{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(k)} - (\mathbf{w}^*)^{(1)}\right)^2\right] &\leq \prod_{j=1}^{T}\left(1-\eta_j\lambda^{(k)}\right)^2 v_0^{(1)} + \lambda^{(k)}\sigma^2\sum_{j=1}^{T}\eta_j^2\prod_{i=j+1}^{T}\left(1-\eta_i\lambda^{(k)}\right)^2 \\
&\leq \exp\left(-2\sum_{j=1}^{T}\eta_j\lambda^{(k)}\right)v_0^{(1)} + \lambda^{(k)}\sigma^2\sum_{j=1}^{T}\eta_j^2\exp\left(-2\sum_{i=j+1}^{T}\eta_i\lambda^{(k)}\right).
\end{aligned}$$

We will show that for every $k$, we have

$$v_T^{(k)} \leq \frac{v_0^{(k)}}{T^3} + \frac{50\log_2\kappa}{\lambda^{(k)}T}\cdot\sigma^2., \tag{D.7}$$

which directly implies the theorem.

We will consider the first $T/3$ steps. The guarantee that we will prove for these iterations is: for any $t \leq A$, $v_t^{(k)} \leq (1-\lambda^{(k)}/R^2)^{2t}v_0^{(k)} + \frac{\sigma^2}{R^2}$.

This can be proved easily by induction. Clearly this is true when $t = 0$. Suppose it is true for $t-1$, let's consider step $t$. By recursion of $v_t^{(k)}$ we know

$$\begin{aligned} v_t^{(k)} &= (1-\lambda^{(k)}/R^2)^2 v_{t-1}^{(k)} + \lambda^{(k)}\sigma^2/(R^2)^2 \\ &\le (1-\lambda^{(k)}/R^2)^{2t} v_0^{(k)} + \frac{\sigma^2}{R^2}\left((1-\lambda^{(k)}/R^2)^2 + \lambda^{(k)}/R^2\right) \\ &\le (1-\lambda^{(k)}/R^2)^{2t} v_0^{(k)} + \frac{\sigma^2}{R^2}. \end{aligned}$$

Here the second step uses induction hypothesis and the third step uses the fact that $(1-x)^2 + x \le 1$ when $x \in [0,1]$. In particular, since $(1-\lambda^{(k)}/R^2)^{2T/3} \le (1-1/\kappa)^{2T/3} \le (1-1/\kappa)^{3\kappa \log T} = 1/T^3$, we know at the end of the first phase, $v_A^{(k)} \le v_0^{(k)}/T^3 + \frac{\sigma^2}{R^2}$.

In the second $T/3$ steps, the guarantee would be: for any $t \le T/3$, $v_{A+t}^{(k)} \le v_0^{(k)}/T^3 + 2\eta_{A+t}\sigma^2$.

We will again prove this by induction. The base case ($t = 0$) follows immediately from the guarantee for the first part. Suppose this is true for $A+t-1$, let us consider $A+t$, again by recursion we know

$$\begin{aligned} v_{A+t}^{(k)} &= (1-\lambda^{(k)}\eta_{A+t-1})^2 v_{A+t-1}^{(k)} + \lambda^{(k)}\sigma^2\eta_{A+t-1}^2 \\ &\le v_0^{(k)}/T^3 + 2\eta_{A+t-1}\sigma^2\left((1-\lambda^{(k)}\eta_{A+t-1})^2 + \frac{1}{2}\lambda^{(k)}\eta_{A+t-1}\right) \\ &\le v_0^{(k)}/T^3 + 2\eta_{A+t-1}\sigma^2(1-\frac{1}{2}\mu\eta_{A+t-1}) \le v_0^{(k)}/T^3 + 2\eta_{A+t}\sigma^2. \end{aligned}$$

Here the last line uses the fact that $2\eta_{A+t-1}(1-\frac{1}{2}\mu\eta_{A+t-1}) \le 2\eta_{A+t}\sigma^2$, which is easy to verify by our choice of $\eta$. Therefore, at the end of the second part, we have $v_B^{(k)} \le v_0^{(k)}/T^3 + \frac{2\sigma^2}{\mu(\kappa+T/6)}$.

Finally we will analyze the third part. Let $\hat{T} = T/3\log_2\kappa$, we will consider the variance $v_{B+\ell\hat{T}}^{(k)}$ at the end of each phase. We will make the following claim by induction:

**Remark 2.** *Suppose* $2^\ell \cdot \mu \le \lambda^{(k)}$, *then*

$$v_{B+\ell\hat{T}}^{(k)} \le v_B^{(k)}\exp(-3\ell) + 2\hat{T}\eta_\ell^2\lambda^{(k)}\sigma^2.$$

*Proof.* We will prove this by induction. When $\ell = 0$, clearly we have $v_B^{(k)} \le v_B^{(k)}$ so the claim is true. Suppose the claim is true for $\ell - 1$, we will consider what happens after the algorithm uses $\eta_\ell$ for $\hat{T}$ steps. By the recursion of the variance we have

$$v_{\ell\hat{T}}^{(k)} \leq v_{(\ell-1)\hat{T}}^{(k)} \cdot \exp(-2\eta_\ell \cdot \lambda^{(k)}\hat{T}) + \hat{T}\eta_\ell^2\lambda^{(k)}\sigma^2.$$

Since $2^\ell \cdot \mu \leq \lambda^{(k)}$, we know $\exp(-2\eta_\ell \cdot \lambda^{(k)}\hat{T}) \leq \exp(-3)$. Therefore by induction hypothesis we have

$$v_{B+\ell\hat{T}}^{(k)} \leq v_B^{(k)} \exp(-3\ell) + \exp(-3) \cdot 2\hat{T}\eta_{\ell-1}^2\lambda^{(k)} + \hat{T}\eta_\ell^2\lambda^{(k)} \leq v_B^{(k)} \exp(-3\ell) + 2\hat{T}\eta_\ell^2\lambda^{(k)}.$$

This finishes the induction. □

By Claim 2, Let $\ell^*$ denote the number satisfying $2^{\ell^*} \cdot \mu \leq \lambda^{(k)} < 2^{\ell^*+1} \cdot \mu$, by this choice we know $\mu/\lambda^{(k)} \geq \frac{1}{2}\exp(-3\ell^\star)$ we have

$$\begin{aligned} v_T^{(k)} &\leq v_{B+\ell^*\hat{T}}^{(k)} \leq v_B^{(k)} \exp(-3\ell^*) + 2\hat{T}\eta_{\ell^*}^2\lambda^{(k)}\sigma^2 \\ &\leq \frac{v_0^{(k)}}{T^3} + \frac{24\sigma^2}{\lambda^{(k)}T} + \frac{50\log_2\kappa}{3\lambda^{(k)}T} \cdot \sigma^2. \\ &\leq \frac{v_0^{(k)}}{T^3} + \frac{50\log_2\kappa}{\lambda^{(k)}T} \cdot \sigma^2. \end{aligned}$$

Therefore, the function value is bounded by $\mathbb{E}\left[f(\mathbf{w}_T^{\text{var}})\right] - f(\mathbf{w}^*) = \sum_{i=1}^d \lambda^{(k)} v_T^{(k)} \leq \frac{50\log_2\kappa}{T} \cdot \sigma^2 d.$ □

*Proof of Proposition 20.* The proof of the proposition works similar to the proof of the strongly convex case of Theorem 19, wherein, we combine the result of Proposition 64 with Lemma 59 and Lemma 55 to obtain the result. □

## *D.4 Proofs of results in Section 6.4.3*

All of our counter-examples in this section are going to be the same simple function. Let the inputs $x$ be such that only a single co-ordinate be active on each example. We refer to this case as the "discrete" case. Furthermore, let each co-ordinate be a Gaussian with mean 0 and variance for the first $d/2$ directions being $d\kappa/3$ and the final $d/2$ directions being 1.

Furthermore, consider the noise to be additive (and independent of $\mathbf{x}$) with mean zero. This indicates that $R^2 = \kappa$ for this problem.

Intuitively, we will show that in order to have a small error in the first eigendirection (with eigenvalue $\kappa$), one need to set a small learning rate $\eta_t$ which would be too small to achieve a small error in the second eigendirection (with eigenvalue 1). As a useful tool, we will decompose the variance in the two directions corresponding to $\kappa$ eigenvalue and 1 eigenvalue respectively as follows:

$$\begin{aligned} v_T^{(1)} \overset{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(1)} - (\mathbf{w}^*)^{(1)}\right)^2\right] &= \prod_{j=1}^{T}(1-\eta_j\kappa)^2 v_0^{(1)} + \kappa\sigma^2 \sum_{j=1}^{T}\eta_j^2 \prod_{i=j+1}^{T}(1-\eta_i\kappa)^2 \\ &\geq \exp\left(-2\sum_{j=1}^{T}\eta_j\kappa\right) v_0^{(1)} + \kappa\sigma^2\sum_{j=1}^{T}\eta_j^2 \exp\left(-2\sum_{i=j+1}^{T}\eta_i\kappa\right) \text{ and} \qquad \text{(D.8)} \\ v_T^{(2)} \overset{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(2)} - (\mathbf{w}^*)^{(2)}\right)^2\right] &= \prod_{j=1}^{T}(1-\eta_j)^2 v_0^{(2)} + \sigma^2 \sum_{j=1}^{T}\eta_j^2 \prod_{i=j+1}^{T}(1-\eta_i)^2 \\ &\geq \exp\left(-2\sum_{j=1}^{T}\eta_j\right) v_0^{(2)} + \sigma^2\sum_{j=1}^{T}\eta_j^2 \exp\left(-2\sum_{i=j+1}^{T}\eta_i\right). \qquad \text{(D.9)} \end{aligned}$$

**Theorem 65.** *Consider the additive noise oracle setting, where, we have access to stochastic gradients satisfying:*

$$\widehat{\nabla} f(\mathbf{w}) = \nabla f(\mathbf{w}) + \zeta = \mathbf{H}(\mathbf{w}-\mathbf{w}^*) + \zeta,$$

*where,*

$$\mathbb{E}\left[\zeta|\mathbf{w}\right] = 0, \textit{ and, } \mathbb{E}\left[\zeta\zeta^\top|\mathbf{w}\right] = \sigma^2\mathbf{H}$$

*There exists a universal constant $C > 0$, and a problem instance, such that for SGD algorithm with any $\eta_t \leq 1/2\kappa$ for all $t$*[1]*, we have*

$$\limsup_{T\to\infty} \frac{\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*)}{(\sigma^2 d/T)} \geq C\frac{\kappa}{\log(\kappa+1)}.$$

[1]Learning rate more than $2/\kappa$ will make the algorithm diverge.

*Proof.* Fix $\tau = \kappa/C \log(\kappa+1)$ where $C$ is a universal constant that we choose later. We need to exhibit that the lim sup is larger than $\tau$. For simplicity we will also round $\kappa$ up to the nearest integer.

Let $T$ be a given number. Our goal is to exhibit a $\tilde{T} > T$ such that $\frac{f(\mathbf{w}_{\tilde{T}})-f(\mathbf{w}^*)}{\left(\sigma^2/\tilde{T}\right)} \geq \tau$. Given the step size sequence $\eta_t$, consider the sequence of numbers $T_0 = T, T_1, \cdots, T_\kappa$ such that $T_i$ is the first number that

$$\frac{1}{\kappa} \leq \sum_{t=T_{i-1}+1}^{T_i} \eta_t \leq \frac{3}{\kappa}.$$

Note that such a number always exists because all the step sizes are at most $2/\kappa$. We will also let $\Delta_i$ be $T_i - T_{i-1}$. Firstly, from (D.8) and (D.9), we see that $\sum_t \eta_t = \infty$. Otherwise, the bias will never decay to zero. If $f(\mathbf{w}_{T_{i-1}+\Delta_i}) - f(\mathbf{w}^*) > \frac{\tau\sigma^2 d}{T_{i-1}+\Delta_i}$ for some $i = 1, \cdots, \kappa$, we are done. If not, we obtain the following relations:

$$\begin{aligned}
&\frac{\sigma^2}{\Delta_1} \leq \sigma^2 \sum_{t=1}^{\Delta_1} \eta_{T_0+t}^2 \leq \frac{\exp(3)}{\kappa} \cdot \mathbb{E}\left[\left(\mathbf{w}_{T_0+\Delta_1}^{(1)} - (\mathbf{w}^*)^{(1)}\right)^2\right] \\
&\leq \exp(3)(f(\mathbf{w}_{T_0+\Delta_1}) - f(\mathbf{w}^*)) \leq \frac{\exp(3)\tau\sigma^2}{T_0+\Delta_1} \\
&\Rightarrow T_0 \leq (\exp(3)\tau - 1)\,\Delta_1.
\end{aligned}$$

Here the second inequality is based on (D.8). We will use $C_1$ to denote $\exp(3)$. Similarly, we have

$$\frac{\sigma^2}{\Delta_2} \leq \sigma^2 \sum_{t=1}^{\Delta_2} \eta_{T_1+t}^2 \leq \frac{C_1}{\kappa}\mathbb{E}\left[\left(\mathbf{w}_{T_1+\Delta_2}^{(1)} - (\mathbf{w}^*)^{(1)}\right)^2\right] \leq C_1(f(\mathbf{w}_{T_1+\Delta_2}) - f(\mathbf{w}^*)) \leq \frac{C_1\tau\sigma^2}{T_1+\Delta_2}$$

$$\Rightarrow T_1 \leq (C_1\tau - 1)\,\Delta_2 \quad \Rightarrow \quad T_0 \leq \frac{(C_1\tau-1)^2}{C_1\tau}\Delta_2.$$

Repeating this argument, we can show that

$$T = T_0 \leq \frac{(C_1\tau-1)^i}{(C_1\tau)^{i-1}}\Delta_i \quad \text{and} \quad T_i \leq \frac{(C_1\tau-1)^{j-i}}{(C_1\tau)^{j-i-1}}\Delta_j \quad \forall\, i<j.$$

We will use $i = 1$ in particular, which specializes to

$$T_1 \leq \frac{(C_1\tau-1)^{j-1}}{(C_1\tau)^{j-2}}\Delta_j \quad \forall\, j \geq 2.$$

Using the above inequality, we can lower bound the sum of $\Delta_j$ as

$$\begin{aligned}\sum_{j=2}^{\kappa} \Delta_j \geq T_1 \cdot \sum_{j=2}^{\kappa} \frac{(C_1\tau)^{j-2}}{(C_1\tau - 1)^{j-1}} \geq T_1 \cdot \frac{1}{C_1\tau} \cdot \sum_{j=2}^{\kappa} \left(1 + \frac{1}{C_1\tau}\right)^{j-2} \\ \geq T_1 \cdot \frac{1}{C_1\tau} \cdot \exp\left(\kappa/(C_1\tau)\right). \qquad \text{(D.10)}\end{aligned}$$

This means that

$$\begin{aligned}\mathbb{E}\left[f(\mathbf{w}_{T_i})\right] - f(\mathbf{w}^*) \geq \frac{d}{2} \cdot \mathbb{E}\left[\left(\mathbf{w}_{T_i}^{(2)} - (\mathbf{w}^*)^{(2)}\right)^2\right] \geq \exp(-6)\sigma^2 d \cdot \sum_{i=1}^{\Delta_1} \eta_{T+i}^2 \\ \geq \frac{\exp(-6)\sigma^2 d}{\Delta_1} \geq \frac{\exp(-6)\sigma^2 d}{T_1} \geq \frac{\exp\left(\kappa/(C_1\tau) - 3\right)}{C_1\tau} \cdot \frac{\sigma^2 d}{\sum_{j=2}^{\kappa} \Delta_j},\end{aligned}$$

where we used (D.10) in the last step. Rearranging, we obtain

$$\frac{\mathbb{E}\left[f(\mathbf{w}_{T_\kappa})\right] - f(\mathbf{w}^*)}{(\sigma^2 d/T_\kappa)} \geq \frac{\exp\left(\kappa/(C_1\tau) - 3\right)}{C_1\tau}.$$

If we choose a large enough $C$ (e.g., $3C_1$), the right hand side is at least $\frac{\exp((C/C_1)\log(\kappa+1)-3)}{\kappa} \geq \kappa$.

□

*Proof of Theorem 21.* Theorem 21 follows as a straightforward consequence of Theorem 65 and Lemma 61. □

## D.5 Details of experimental setup

### D.5.1 Synthetic 2-d Quadratic Experiments

As mentioned in the main chapter, we consider four condition numbers namely $\kappa \in \{10, 50, 250, 1250\}$. We run all experiments for a total of $25 \times \kappa_{\max} = 25 \times 1250$ iterations. The two eigenvalues of the Hessian are $\kappa$ and 1 respectively, and noise level is $\sigma^2 = 1$ and we average our results with ten random seeds. All our grid search results are conducted on a $8 \times 8$ grid of learning rates $\times$ decay factor and whenever a best run lands at the edge of the grid, the grid is extended so that we have the best run in the interior of the grid search.

For the $O(1/t)$ learning rate, we search for decay parameter over $8-$points log-spaced between $\{1/(500\kappa), 500000/\kappa\}$. The starting learning rate is searched over 8 points logarithmically spaced between $\{0.005/(\kappa), 50000.0/\kappa\}$.

For the $O(1/\sqrt{t})$ learning rate, the decay parameter is searched over 8 logarithmically spaced points between $\{1/(100\kappa), 10000000/\kappa\}$. The starting learning rate is searched between $\{1/50000, 100\}$ with 8 logarithmically spaced points.

For the exponential learning rate schemes, the decay parameter is searched between $\{1/(8000 * \kappa), 100/\kappa\}$. The learning rate is searched between $\{1/100000, 0.5\}$.

### *D.5.2 Non-Convex experiments on cifar-10 dataset with a 44-layer residual net*

As mentioned in the main chapter, for all the experiments, we use the Nesterov's Accelerated gradient method [85] implemented in pytorch [2] with a momentum set to 0.9 and batchsize set to 128, total number of training epochs set to 100, $\ell_2$ regularization set to 0.0005.

With regards to learning rates, we consider $10-$values geometrically spaced as $\{1, 0.6, \cdots, 0.01\}$. To set the decay factor for any of the schemes such as 6.5,6.6, and 6.7, we use the following rule. Suppose we have a desired learning rate that we wish to use towards the end of the optimization (say, something that is 100 times lower than the starting learning rate, which is a reasonable estimate of what is typically employed in practice), this can be used to obtain a decay factor for the corresponding decay scheme. In our case, we found it advantageous to use an additively spaced grid for the learning rate $\gamma_t$, i.e., one which is searched over a range $\{0.0001, 0.0002, \cdots, 0.0009, 0.001, \cdots, 0.009\}$ at the $80^{th}$ epoch, and cap off the minimum possible learning rate to be used to be 0.0001 to ensure that there is progress made by the optimization routine. For any of the experiments that yield the best performing gridsearch parameter that falls at the edge of the grid, we extend the grid to ensure that the finally chosen hyperparameter lies in the interior of the grid. All our gridsearches are run such that we separate a tenth of the training dataset as a validation set and

[2] `https://github.com/pytorch`

train on the remaining $9/10^{th}$ dataset. Once the best grid search parameter is chosen, we train on the entire training dataset and evaluate on the test dataset and present the result of the final model (instead of choosing the best possible model found during the course of optimization).

# VITA

Rahul Kidambi was born in Tiruchirappalli, a city that is home to the wonderful Cauvery River in the state of Tamil Nadu, India. He is currently a PhD candidate under the supervision of Prof. Sham M. Kakade at the University of Washington, Seattle. His interest is in the practical development of well-performing intelligent systems that admit a reasonable degree of robustness and efficiency.

During his PhD, Rahul spent a summer interning at Microsoft Research India with Praneeth Netrapalli and Prateek Jain. Prior to his PhD, he spent time (again) at Microsoft Research India, working with Sundararajan Sellamanickam and S. Sathiya Keerthi. He holds a Bachelors degree in Electronics and Communication Engineering at National Institute of Technology Tiruchirappalli (where he graduated as the best outgoing student of his department), and a Masters degree in Electrical and Computer Engineering from University of California Santa Barbara.

Aside from research, Rahul loves travelling, running, swimming, ping pong, reading fiction and spending time with his family.