Scalable Machine Learning Algorithms for Biological Sequence Data

Jeffrey Chan

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-108
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-108.html

May 14, 2021

Scalable Machine Learning Algorithms for Biological Sequence Data

by

Jeffrey Chan

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Yun S. Song, Chair
Professor Ian Holmes
Professor Jennifer Listgarten

Spring 2021

Scalable Machine Learning Algorithms for Biological Sequence Data

Abstract

Scalable Machine Learning Algorithms for Biological Sequence Data

by

Jeffrey Chan

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Yun S. Song, Chair

Recent advances in sequencing and synthesis technologies have sparked extraordinary growth in large-scale biological experimentation and data collection. This explosive growth necessitates the development of scalable yet accurate methods to investigate increasingly complex biological questions. Machine learning has become a vital tool for addressing the needs of computational biology blending complex statistical models with efficient computation to uncover the underpinnings of biology.

In this dissertation, I develop three novel machine learning algorithms tailored towards biological sequence data to aid in answering such biological questions. The first method is a general-purpose statistical framework for inference of population genetic parameters. Previous methods focused on developing model approximation methods for a restricted class of models or reducing datasets to a set of hand-crafted summary statistics and comparing them against simulated data. Our framework uses a exchangeable neural network which respects the permutation-invariant symmetries of the data to learn the mapping from simulated datasets to the population genetic parameters of interest.

The second method extends the ideas from the first method to a more challenging setting where segmentation of the genotypes is necessary to determine tracts of archaic admixture. In this setting, the data are permutation-equivariant requiring a neural network architecture that results in accurate segmentation of archaic admixture tracts.

Finally, the third method focuses on the problem of search in protein engineering to discover high fitness protein sequences of interest. Standard bandit optimization methods often focus on experimental feedback that is purely sequential. In protein engineering, advances in high-throughput synthesis and experimentation can often lead to large batches of size as large as 10^5 where the size of the batch can often be much larger than the number of rounds of experimentation. We propose a family of parallel contextual linear bandit algorithms and analyze their regret bounds.

To my parents.

Contents

List of Figures

Acknowledgments

First and foremost, I would like to express my sincere and utmost gratitude for my advisor Yun Song. His technical expertise combined with his "kid in a candy store" curiosity served as a great role model from the first day I stepped into his office during undergrad. With his support and encouragement, I was provided the freedom to explore my interests and forge my own path. I often sought his guidance on navigating whichever research obstacle lay before me with the knowledge and comfort that he works tirelessly for his students and always has their best interests at heart.

I have been fortunate to have many outstanding collaborators: Sara Mathieson, Jeff Spence, Valerio Perrone, Paul Jenkins, Aldo Pacchiano, and Nilesh Tripuraneni. Each of them were a joy to work with and constantly taught me new things. In addition, I am extremely grateful for my labmates throughout the years: Alan Aw, Sanjit Batra, Gonzalo Benegas, Nick Bhattacharya, Khanh Dao Duc, Yun Deng, Dan Erdmann-Pham, Jonathan Fischer, Geno Guerra, Ethan Jewett, Jack Kamm, Antoine Koehl, Joyce Liu, Shishi Luo, Sebastian Prillo, Zvi Rosen, Jeff Spence, Jonathan Terhorst, Neil Thomas, Miaoyan Wang, Yutong Wang, Alex Whatley, and Jane Yu. I could not have asked for a more intelligent and kind group of people to learn from, bounce ideas with, and joke around with. Each and everyone of them was instrumental to my graduate school experience. In particular, I am incredibly indebted to Jeff Spence whose mentorship and friendship always gave me new perspectives, research insights, and a warm "Hey Jeffers" greeting every morning.

I was warned early on that graduate school can be a rather isolating experience and that it was vital for me to have a support system to share the ups and downs with. I am fortunate enough to have an amazing group of friends at Berkeley who have always been steady and present to support me throughout my time here. I am deeply thankful for and indebted to Isabella Huang for welcoming me with open arms into her makeshift family of Canadians and for making sure that above all else life always stays fun. To that crew of friends —Isabella, Judy Shen, Samantha Huang, Ivan Lee, and Eleanor Siow —thank you for making adventures around the Bay Area full of laughs, good food, and inexplicably fishing. You made Berkeley feel like home for the past few years. I would also like to thank my fellow EECS graduate student, Nilesh Tripuraneni for always being down to grab food at a moment's notice and reGhale me in his latest stories. I am also thankful to have met Libby Kao who through her incredible kindness, empathy, and sense of humor manages to make even the hardest days feel easier. Though I created a great support system at Berkeley, longtime friends from high school, summer camps, and MIT reminded me of all of the facets of life outside of academia regardless of whether they were just across the Bay or halfway around the world. For that, I thank Jean Shiao, Kevin Tian, Diana Cai, Fan-Hal Koung, Joseph Kim, Grace Shin, Riana LoBu, Carmen Ng, Jorge Perez, Alex Lesman, Sidhant Pai, Alex Jaffe, and Robert Brik. Each of whom offered their own unique combination of laughs, support, and counsel.

Lastly, I thank my parents who provided me with the financial stability and relentless encouragement to pursue my career and intellectual interests. Their all-encompassing com-

mitment to my education and learning ranged from driving me to classes and extracurriculars to the more unorthodox quizzing me on a barrage of math puzzles when they just wanted me to stop bickering with my sister as a child. Their support and devotion made this all possible and to them I dedicate this dissertation.

Chapter 1

Introduction

1.1 Background

Advances in sequencing technologies and synthetic biology experimental techniques have led to an explosion of biological sequence data over the last decade allowing biologists to investigate scientific questions with much greater resolution than was previously possible. In parallel, tremendous progress in machine learning and high-performance computing over the past decade has paved the way for complex analyses of large biological sequence datasets to assist in answering these scientific questions.

Blending together out-of-the-box machine learning techniques with modern datasets can often lead to fruitful results [6]. In these settings, successes in applications such as image processing and natural language processing can be immediately translated to questions in biology. This has led to a rush of progress in longstanding computational biology problems. However, occasionally the biological problem setting of interest requires tailor-made methodological approaches wielding the insights and ideas from the machine learning community while wrestling often messy biological constraints. Here we focus on problem settings in two fields that require such tailor-made approaches: population genetics and protein engineering.

Realistic population genetic models typically underpinned by combinatorial stochastic processes yield a simple generative process, but often lead to intractable exact inference outside of a small subset of the model class. One approach taken by population geneticists is to make approximations that ease inference while still capturing the complexities of the data [80, 58]. However, this is rather time consuming and not necessarily applicable to all settings of interest leaving the opportunity ripe to combine ideas from Bayesian inference and machine learning towards a generalizable black-box approach.

The field of protein engineering centers on a search problem where one iteratively performs a set of experiments to determine the fitness of a particular batch of proteins seeking a diverse set of high fitness proteins. This can be naturally cast as an iterative decision making problem. While standard iterative decision making algorithms are sequential, advances in synthetic biology allowing for parallel experimentation require batch iterative decision

making algorithms.

1.2 Overview

In this dissertation, I devise machine learning methods to address three biological problems where biological constraints necessitated the development of new methods.

In Chapter 2, I present a framework for population genetic inference that is particularly useful for complex models not amenable to tractable inference using the standard statistical toolkit. In such models, the computation of the likelihood even pointwise can be intractable. Instead, I take the approach of using population genetics simulators to generate data and train a deep neural network to learn the mapping from observed data to posteriors over population genetic parameters. In particular, we develop an architecture that respects the permutation invariant structure of population genetic data to learn this mapping. This framework can be applied in a black-box fashion across a variety of simulation-based tasks, both within and outside biology. We demonstrate the power of our approach on the recombination hotspot testing problem, outperforming the state-of-the-art.

In Chapter 3, I tackle the problem of identifying genetic variants in the human genome derived from interbreeding with "ghost" (unknown) archaic hominids which sheds light on how humans adapted to past environmental changes. Previous methods lack the statistical power to confidently infer the presence of archaic admixture. I develop a flexible reference-free Bayesian inference method which can elegantly incorporate any known population genetic information while remaining flexible and robust to uncertainty. Our method proposes a permutation-equivariant neural network tailored towards population genetic data and applies it both to classification and to segmentation (also known as chromosome painting) of archaic DNA. We significantly outperform the state-of-the-art for reference-free classification and segmentation of archaic DNA and comparable performance with existing reference-dependent methods.

In Chapter 4, I present a new class of linear bandit algorithms along with accompanying analysis for the problem of exploration in protein engineering applications. Standard bandit algorithms are typically designed for sequential decision-making under uncertainty. However, in protein engineering vast levels of parallelism can be performed with as many as 10^4 protein sequences being evaluated at once. I present a family of parallelized linear bandit algorithms applicable to the problem of protein engineering. I then provide analyses to understand how underlying parameters scale as parallelism and time are varied. Finally, I demonstrate the efficacy of our suite of algorithms on a variety of synthetic and real world datasets.

Chapter 2

Exchangable Neural Networks for Population Genetic Inference

This is joint work with Valerio Perrone, Jeffrey P. Spence, Paul A. Jenkins, Sara Mathieson, and Yun S. Song. Ben Graham and Yuval Simons provided helpful discussions. This was first published in *NeurIPS* [21].

2.1 Introduction

Statistical inference in population genetics aims to quantify the evolutionary events and parameters that led to the genetic diversity we observe today. Population genetic models are typically based on the coalescent [51], a stochastic process describing the distribution over genealogies of a random exchangeable set of DNA sequences from a large population. Inference in such complex models is challenging. First, standard coalescent-based likelihoods require integrating over a large set of correlated, high-dimensional combinatorial objects, rendering classical inference techniques inapplicable. Instead, likelihoods are implicitly defined via scientific simulators (i.e., generative models), which draw a sample of correlated trees and then model mutation as Poisson point processes on the sampled trees to generate sequences at the leaves. Second, inference demands careful treatment of the exchangeable structure of the data (a set of sequences), as disregarding it leads to an exponential increase in the already high-dimensional state space.

Current likelihood-free methods in population genetics leverage scientific simulators to perform inference, handling the exchangeable-structured data by reducing it to a suite of low-dimensional, permutation-invariant summary statistics [13, 79]. However, these hand-engineered statistics typically are not statistically sufficient for the parameter of interest. Instead, they are often based on the intuition of the user, need to be modified for each new task, and are not amenable to hyperparameter optimization strategies since the quality of the approximation is unknown.

The goal of this work is to develop a general-purpose inference framework for raw pop-

ulation genetic data that is not only likelihood-free, but also summary statistic-free. We achieve this by designing a neural network that exploits data exchangeability to learn functions that accurately approximate the posterior. While deep learning offers the possibility to work directly with genomic sequence data, poorly calibrated posteriors have limited its adoption in scientific disciplines [33]. We overcome this challenge with a training paradigm that leverages scientific simulators and repeatedly draws fresh samples at each training step. We show that this yields calibrated posteriors and argue that, under a likelihood-free inference setting, deep learning coupled with this 'simulation-on-the-fly' training has many advantages over the more commonly used Approximate Bayesian Computation (ABC) [13, 66]. To our knowledge, this is the first method that handles the raw exchangeable data in a likelihood-free context.

As a concrete example, we focus on the problems of recombination hotspot testing and estimation. Recombination is a biological process of fundamental importance, in which the reciprocal exchange of DNA during cell division creates new child gamete chromosomes that are a mosaic of the two parental chromosomes. From an evolutionary point of view, an important consequence is that different positions on the genome have different genealogical histories. Experiments have shown that many species exhibit *recombination hotspots*, i.e., short segments of the genome with high recombination rates [64] leading to extremely decorrelated genealogies between the flanking regions of the hotspot. The task of recombination hotspot testing is to predict the location of recombination hotspots given genetic polymorphism data. Accurately localizing recombination hotspots would illuminate the biological mechanism that underlies recombination, and could help geneticists map the mutations causing genetic diseases [38]. We demonstrate through experiments that our proposed framework outperforms the state-of-the-art on the hotspot detection problem.

Our main contributions are:

- A novel exchangeable neural network that respects permutation invariance and maps from the data to the posterior distribution over the parameter of interest.
- A simulation-on-the-fly training paradigm, which leverages scientific simulators to achieve calibrated posteriors.
- A general-purpose likelihood-free Bayesian inference method that combines the exchangeable neural network and simulation-on-the-fly training paradigm to both discrete and continuous settings. Our method can be applied to many population genetic settings by making straightforward modifications to the simulator and the prior, including demographic model selection, archaic admixture detection, and classifying modes of natural selection.
- An application to a single-population model for recombination hotspot testing and estimation, outperforming the model-based state-of-the-art, `LDhot`. Our approach can be seamlessly extended to more complex model classes, unlike `LDhot` and other model-based methods.

Our software package `defiNETti` is publicly available at `https://github.com/popgenmethods/defiNETti`.

2.2 Related Work

Likelihood-free methods like ABC have been widely used in population genetics [13, 66, 17, 97, 84]. In ABC the parameter of interest is simulated from its prior distribution, and data are subsequently simulated from the generative model and reduced to a pre-chosen set of summary statistics. These statistics are compared to the summary statistics of the real data, and the simulated parameter is weighted according to the similarity of the statistics to derive an empirical estimate of the posterior distribution. However, choosing summary statistics for ABC is challenging because there is a trade-off between loss of sufficiency and computational tractability. In addition, there is no direct way to evaluate the accuracy of the approximation.

Other likelihood-free approaches have emerged from the machine learning community and have been applied to population genetics, such as support vector machines (SVMs) [76, 63], single-layer neural networks [16], and deep learning [79]. Recently, a (non-exchangeable) convolutional neural network method was proposed for raw population genetic data [29]. The connection between likelihood-free Bayesian inference and neural networks has also been studied previously [43, 62]. An attractive property of these methods is that, unlike ABC, they can be applied to multiple datasets without repeating the training process (i.e., amortized inference). However, current practice in population genetics collapses the data to a set of summary statistics before passing it through the machine learning models. Therefore, the performance still rests on the ability to laboriously hand-engineer informative statistics, and must be repeated from scratch for each new problem setting.

The inferential accuracy and scalability of these methods can be improved by exploiting symmetries in the input data. Permutation-invariant models have been previously studied in machine learning for SVMs [81] and recently gained a surge of interest in the deep learning literature. Recent work on designing architectures for exchangeable data include [69], [34], and [98], which exploit parameter sharing to encode invariances.

We demonstrate these ideas on the discrete and continuous problems of recombination hotspot testing and estimation, respectively. To this end, several methods have been developed (see, e.g., [27, 59, 95] for the hotspot testing problem). However, none of these are scalable to the whole genome, with the exception of `LDhot` [9, 94], so we limit our comparison to this latter method. `LDhot` relies on a composite likelihood, which can be seen as an approximate likelihood for summaries of the data. It can be computed only for a restricted set of models (i.e., an unstructured population with piecewise constant population size), is unable to capture dependencies beyond those summaries, and scales at least cubically with the number of DNA sequences. The method we propose in this paper scales linearly in the number of sequences while using raw genetic data directly.

LDhot details

The most widely-used technique for recombination hotspot testing is `LDhot` as described in [9]. The method performs a generalized composite likelihood ratio test using the two-locus composite likelihood based on [41] and [61]. The composite two-locus likelihood approximates the joint likelihood of a window of SNPs w by a product of pairwise likelihoods

$$CL(\rho \mid \mathbf{x}) = \prod_{1 \leq |i-j| \leq z} L(\rho_{ij} \mid \mathbf{x}_{ij}),$$

where X_{ij} denotes the data restricted only to SNPs i and j, and ρ_{ij} denotes the recombination rate between those sites. Only SNPs within some distance, say $z = 50$, are considered.

Two-locus likelihoods are computed via an importance sampling scheme under a constant population size ($\eta = 1$) as in [61]. The likelihood ratio test uses a null model of a constant recombination rate and an alternative model of a differing recombination rate in the center of the window under consideration:

$$\Lambda = -2 \log \left(\frac{\sup_{\rho_{\text{hot}}, \rho_{\text{bg}}} CL(\rho_{\text{hot}}, \rho_{\text{bg}} \mid X)}{\sup_{\rho_{\text{const}}} CL(\rho_{\text{const}} \mid X)} \right).$$

The two-locus likelihood can only be applied to a single population with constant population size, constant mutation rate, and without natural selection. Furthermore, the two-locus likelihood is an uncalibrated approximation of the true joint likelihood. In addition, [94] and [9] performed simulation studies showing that LDhot has good power but their simulation scenarios were unrealistic because its null hypothesis leads to a comparison against a biologically unrealistic flat background rate. In order to fairly compare our likelihood-free approach against the composite likelihood-based method in realistic human settings, we extended the `LDhot` methodology to apply to a piecewise constant population sizes using two-locus likelihoods computed by the software `LDpop` [46]. Unlike the method described in [94], our implementation of `LDhot` uses windows defined in terms of SNPs rather than physical distance in order to measure accuracy via ROC curves, since the likelihood ratio test is a function of number of SNPs. Note that computing the approximate two-locus likelihoods for a grid of recombination values is at least $O(n^3)$, which could be prohibitive for large sample sizes.

2.3 Methods

Problem Setup

Likelihood-free methods use coalescent simulators to draw parameters from the prior $\theta^{(i)} \sim \pi(\theta)$ and then simulate data according to the coalescent $\mathbf{x}^{(i)} \sim \mathbb{P}(\mathbf{x} \mid \theta^{(i)})$, where i is the index of each simulated dataset. Each population genetic datapoint $\mathbf{x}^{(i)} \in \{0, 1\}^{n \times d}$ typically takes the form of a binary matrix, where rows correspond to individuals and columns indicate the

presence of a Single Nucleotide Polymorphism (SNP), a variable site in a DNA sequence[1]. Our goal is to learn the posterior $\mathbb{P}(\theta \mid \mathbf{x}_{obs})$, where θ is the parameter of interest and $\mathbf{x}_{obs}$ is the observed data. For unstructured populations the order of individuals carries no information, hence the rows are exchangeable. More concretely, given data $\mathbf{X} = (\mathbf{x}^{(1)}, \dots \mathbf{x}^{(N)})$ where $\mathbf{x}^{(i)} := (x_1^{(i)}, \dots, x_n^{(i)}) \sim \mathbb{P}(\mathbf{x} \mid \theta^{(i)})$ and $x_j^{(i)} \in \{0,1\}^d$, we call $\mathbf{X}$ *exchangeably-structured* if, for every i, the distribution over the rows of a single datapoint is permutation-invariant

$$\mathbb{P}\Big(x_1^{(i)}, \dots, x_n^{(i)} \mid \theta^{(i)}\Big) = \mathbb{P}\Big(x_{\sigma(1)}^{(i)}, \dots, x_{\sigma(n)}^{(i)} \mid \theta^{(i)}\Big),$$

for all permutations σ of the indices $\{1, \dots, n\}$. For inference, we propose iterating the following algorithm.

1. *Simulation-on-the-fly*: Sample a fresh minibatch of $\theta^{(i)}$ and $\mathbf{x}^{(i)}$ from the prior and coalescent simulator.

2. *Exchangeable neural network*: Learn the posterior $\mathbb{P}(\theta^{(i)} \mid \mathbf{x}^{(i)})$ via an exchangeable mapping with $\mathbf{x}^{(i)}$ as the input and $\theta^{(i)}$ as the label.

This framework can then be applied to learn the posterior of the evolutionary model parameters given $\mathbf{x}_{obs}$. The details on the two building blocks of our method, namely the exchangeable neural network and the simulation-on-the-fly paradigm, are given in Section 2.3 and 2.3, respectively.

Exchangeable Neural Network

The goal of the exchangeable neural network is to learn the function $f : \{0,1\}^{n \times d} \to \mathcal{P}_\Theta$, where Θ is the space of all parameters θ and $\mathcal{P}_\Theta$ is the space of all probability distributions on Θ. We parameterize the exchangeable neural network by applying the same function to each row of the binary matrix, then applying a symmetric function to the output of each row, finally followed by yet another function mapping from the output of the symmetric function to a posterior distribution. More concretely,

$$f(\mathbf{x}) := (h \circ g)\big(\Phi(x_1), \dots, \Phi(x_n)\big),$$

where $\Phi : \{0,1\}^d \to \mathbb{R}^{d_1}$ is a function parameterized by a convolutional neural network, $g : \mathbb{R}^{n \times d_1} \to \mathbb{R}^{d_2}$ is a symmetric function, and $h : \mathbb{R}^{d_2} \to \mathcal{P}_\Theta$ is a function parameterized by a fully connected neural network. A variant of this representation is proposed by [69] and [98]. See Figure 2.1 for an example. Throughout the paper, we choose g to be the mean of the element-wise top decile, such that $d_1 = d_2$ in order to allow for our method to be robust to changes in n at test time. Many other symmetric functions such as the element-wise sum, element-wise max, lexicographical sort, or higher-order moments can be employed.

[1]Sites that have > 2 bases are rare and typically removed. Thus, a binary encoding can be used.

This exchangeable neural network has many advantages. While it could be argued that flexible machine learning models could learn the structured exchangeability of the data, encoding exchangeability explicitly allows for faster per-iteration computation and improved learning efficiency, since data augmentation for exchangeability scales as $O(n!)$. Enforcing exchangeability implicitly reduces the size of the input space from $\{0,1\}^{n\times d}$ to the quotient space $\{0,1\}^{n\times d}/S_n$, where S_n is the symmetric group on n elements. A factorial reduction in input size leads to much more tractable inference for large n. In addition, choices of g where d_2 is independent of n (e.g., quantile operations with output dimension independent of n) allows for an inference procedure which is robust to differing number of exchangeable variables between train and test time. This property is particularly desirable for performing inference with missing data.

Simulation-on-the-fly

Supervised learning methods traditionally use a fixed training set and make multiple passes over the data until convergence. This training paradigm typically can lead to a few issues: poorly calibrated posteriors and overfitting. While the latter has largely been tackled by regularization methods and large datasets, the former has not been sufficiently addressed. We say a posterior is calibrated if for $X_{q,A} := \{\mathbf{x} \mid \hat{p}(\theta \in A \mid \mathbf{x}) = q\}$, we have $\mathbb{E}_{\mathbf{x}\in X_{q,A}}[p(\theta \in A \mid \mathbf{x})] = q$ for all q, A. Poorly calibrated posteriors are particularly an issue in scientific disciplines as scientists often demand methods with calibrated uncertainty estimates in order to measure the confidence behind new scientific discoveries (often leading to reliance on traditional methods with asymptotic guarantees such as MCMC).

When we have access to scientific simulators, the amount of training data available is limited only by the amount of compute time available for simulation, so we propose simulating each training datapoint afresh such that there is exactly one epoch over the training data (i.e., no training point is passed through the neural network more than once). We refer to this as *simulation-on-the-fly*. Note that this can be relaxed to pass each training point a small constant number of times in the case of computational constraints on the simulator. This approach guarantees properly calibrated posteriors and obviates the need for regularization techniques to address overfitting. Below we justify these properties through the lens of statistical decision theory.

More formally, define the Bayes risk for prior $\pi(\theta)$ as $R^*_\pi = \inf_T \mathbb{E}_{\mathbf{x}}\mathbb{E}_{\theta\sim\pi}[l(\theta, T(\mathbf{x})]$, with l being the loss function and T an estimator. The excess risk over the Bayes risk resulting from an algorithm A with model class $\mathcal{F}$ can be decomposed as

$$R_\pi(\tilde{f}_A) \;-\; R^*_\pi \;\;=\;\; \underbrace{\Big(R_\pi(\tilde{f}_A) - R_\pi(\hat{f})\Big)}_{\text{optimization error}} + \underbrace{\Big(R_\pi(\hat{f}) - \inf_{f\in\mathcal{F}} R_\pi(f)\Big)}_{\text{estimation error}} + \underbrace{\Big(\inf_{f\in\mathcal{F}} R_\pi(f) - R^*_\pi\Big)}_{\text{approximation error}},$$

where $\tilde{f}_A$ and $\hat{f}$ are the function obtained via algorithm A and the empirical risk minimizer, respectively. The terms on the right hand side are referred to as the optimization, estimation, and approximation errors, respectively. Often the goal of statistical decision theory

is to minimize the excess risk motivating algorithmic choices to control the three sources of error. For example, with supervised learning, overfitting is a result of large estimation error. Typically, for a sufficiently expressive neural network optimized via stochastic optimization techniques, the excess risk is dominated by optimization and estimation errors. Simulation-on-the-fly guarantees that the estimation error is small, and as neural networks typically have small approximation error, we can conclude that the main source of error remaining is the optimization error. It has been shown that smooth population risk surfaces can induce jagged empirical risk surfaces with many local minima [19, 44]. We confirmed this phenomenon empirically in the population genetic setting(Section 2.5) showing that the risk surface is much smoother in the on-the-fly setting than the fixed training setting. This reduces the number of poor local minima and, consequently, the optimization error. The estimator corresponding to the Bayes risk (for the cross-entropy or KL-divergence loss function) is the posterior. Thus, the simulation-on-the-fly training paradigm guarantees generalization and calibrated posteriors (assuming small optimization error).

2.4 Statistical Properties

The most widely-used likelihood-free inference method is ABC. In this section we briefly review ABC and show that our method exhibits the same theoretical guarantees together with a set of additional desirable properties.

Properties of ABC Let $\mathbf{x}_{obs}$ be the observed dataset, S be the summary statistic, and d be a distance metric. The algorithm for vanilla rejection ABC is as follows. Denoting by i each simulated dataset, for $i = 1 \dots N$,

1. Simulate $\theta^{(i)} \sim \pi(\theta)$ and $\mathbf{x}^{(i)} \sim \mathbb{P}(\mathbf{x} \mid \theta^{(i)})$
2. Keep $\theta^{(i)}$ if $d(S(\mathbf{x}^{(i)}), S(\mathbf{x}_{obs})) \leq \epsilon$.

The output provides an empirical estimate of the posterior. Two key results regarding ABC make it an attractive method for Bayesian inference: (1) **Asymptotic guarantee:** As $\epsilon \to 0$, $N \to \infty$, and if S is sufficient, the estimated posterior converges to the true posterior (2) **Calibration of ABC:** A variant of ABC (noisy ABC in [28]) which injects noise into the summary statistic function is calibrated. For detailed proofs as well as more sophisticated variants, see [28]. Note that ABC is notoriously difficult to perform diagnostics on without the ground truth posterior as many factors could contribute to a poor posterior approximation: poor choice of summary statistics, incorrect distance metric, insufficient number of samples, or large ϵ.

Properties of Our Method Our method matches both theoretical guarantees of ABC — (1) asymptotics and (2) calibration — while also exhibiting additional properties: (3)

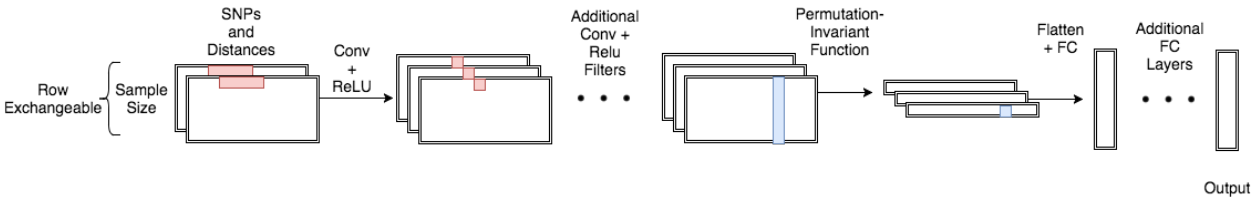

Figure 2.1: A cartoon schematic of the exchangeable architecture for population genetics.

amortized inference, (4) no dependence on user-defined summary statistics, and (5) straightforward diagnostics. While the independence of summary statistics and calibration are theoretically justified in Section 2.3 and 2.3, we provide some results that justify the asymptotics, amortized inference, and diagnostics.

In the simulation-on-the-fly setting, convergence to a global minimum implies that a sufficiently large neural network architecture represents the true posterior within ϵ-error in the following sense: for any fixed error ϵ, there exist H_0 and N_0 such that the trained neural network produces a posterior which satisfies

$$\min_{\mathbf{w}} \mathbb{E}_{\mathbf{x}}\Big[KL\Big(\mathbb{P}(\theta \mid \mathbf{x}) \;\Big\|\; \mathbb{P}_{DL}^{(N)}(\theta \mid \mathbf{x}; \mathbf{w}, H)\Big)\Big] < \epsilon, \tag{2.1}$$

for all $H > H_0$ and $N > N_0$, where H is the minimum number of hidden units across all neural network layers, N is the number of training points, $\mathbf{w}$ the weights parameterizing the network, and KL the Kullback–Leibler divergence between the population risk and the risk of the neural network. Under these assumptions, the following proposition holds.

Proposition 1. *For any* $\mathbf{x}$, $\epsilon > 0$, *and fixed error* $\delta > 0$, *there exists an* $H > H_0$, *and* $N > N_0$ *such that,*

$$KL\Big(\mathbb{P}(\theta \mid \mathbf{x}) \;\Big\|\; \mathbb{P}_{DL}^{(N)}(\theta \mid \mathbf{x}; \mathbf{w}^*, H)\Big) < \delta \tag{2.2}$$

with probability at least $1 - \frac{\epsilon}{\delta}$, *where* $\mathbf{w}^*$ *is the minimizer of* (2.1).

We can get stronger guarantees in the discrete setting common to population genetic data.

Corollary 1. *Under the same conditions, if* $\mathbf{x}$ *is discrete and* $\mathbb{P}(\mathbf{x}) > 0$ *for all* $\mathbf{x}$, *the KL divergence appearing in* (2.2) *converges to* 0 *uniformly in* $\mathbf{x}$, *as* $H, N \to \infty$.

The proofs are given in Section 2.7. These results exhibit both the asymptotic guarantees of our method and show that such guarantees hold for all $\mathbf{x}$ (i.e. amortized inference). Diagnostics for the quality of the approximation can be performed via hyperparameter optimization to compare the relative loss of the neural network under a variety of optimization and architecture settings.

2.5 Empirical Study: Recombination Hotspot Testing

In this section, we study the accuracy of our framework to test for recombination hotspots. As very few hotspots have been experimentally validated, we primarily evaluate our method on simulated data, with parameters set to match human data. The presence of ground truth allows us to benchmark our method and compare against `LDhot`. For the posterior in this classification task (hotspot or not), we use the softmax probabilities. Unless otherwise specified, for all experiments we use the mutation rate, $\mu = 1.1 \times 10^{-8}$ per generation per nucleotide, convolution patch length of 5 SNPs, 32 and 64 convolution filters for the first two convolution layers, 128 hidden units for both fully connected layers, and 20-SNP length windows. The experiments comparing against `LDhot` used sample size $n = 64$ to construct lookup tables for `LDhot` quickly. All other experiments use $n = 198$, matching the size of the CEU population (i.e., Utah Residents with Northern and Western European ancestry) in the 1000 Genomes dataset. All simulations were performed using `msprime` [49]. Gradient updates were performed using Adam [50] with learning rate $1 \times 10^{-3} \times 0.9^{b/10000}$, b being the batch count. In addition, we augment the binary matrix, $\mathbf{x}$, to include the distance information between neighboring SNPs in an additional channel resulting in a tensor of size $n \times d \times 2$.

Simulation Details

We encode population genetic data $\mathbf{x}$ as follows. Let $\mathbf{x}_S$ be the binary $n \times d$ matrix with 0 and 1 as the common and rare nucleotide variant, respectively, where n is the number of sequences, and d is the number of SNPs. Let $\mathbf{x}_D$ be the $n \times d$ matrix storing the distances between neighboring SNPs, so each row of $\mathbf{x}_D$ is identical and the rightmost distance is set to 0. Define $\mathbf{x}$ as the $n \times d \times 2$ tensor obtained by stacking $\mathbf{x}_S$ and $\mathbf{x}_D$. To improve the conditioning of the optimization problem, the distances are normalized such that they are on the order of $[0, 1]$.

The standard generative model for such data is the coalescent, a stochastic process describing the distribution over genealogies relating samples from a population of individuals. The coalescent with recombination [32, 40] extends this model to describe the joint distribution of genealogies along the chromosome. The recombination rate between two DNA locations tunes the correlation between their corresponding genealogies. Population genetic data derived from the coalescent obeys translation invariance along a sequence conditioned on local recombination and mutation rates which are also translation invariant. In order to take full advantage of parameter sharing, our chosen architecture is given by a convolutional neural network with tied weights for each row preceding the exchangeable layer, which is in turn followed by a fully connected neural network.

Recombination Hotspot Details

Recombination hotspots are short regions of the genome with high recombination rate relative to the background. As the recombination rate between two DNA locations tunes the correlation between their corresponding genealogies, hotspots play an important role in complex disease inheritance patterns. In order to develop accurate methodology, a precise mathematical definition of a hotspot needs to be specified in accordance with the signatures of biological interest. We use the following:

Definition 1 (Recombination Hotspot)**.** Let a window over the genome be subdivided into three subwindows $w = (w_l, w_h, w_r)$ with physical distances (i.e., window widths) α_l, α_h, and α_r, respectively, where $w_l, w_h, w_r \in \mathcal{G}$ where $\mathcal{G}$ is the space over all possible subwindows of the genome. Let a mean recombination map $R : \mathcal{G} \to \mathbb{R}_+$ be a function that maps from a subwindow of the genome to the mean recombination rate per base pair in the subwindow. A recombination hotspot for a given mean recombination map R is a window w which satisfies the following properties:

1. Elevated local recombination rate: $R(w_h) > k \cdot \max\big(R(w_l), R(w_r)\big)$
2. Large absolute recombination rate: $R(w_h) > k\tilde{r}$

where $\tilde{r}$ is the median (at a per base pair level) genome-wide recombination rate, and $k > 1$ is the relative hotspot intensity.

The first property is necessary to enforce the locality of hotspots and rule out large regions of high recombination rate, which are typically not considered hotspots by biologists. The second property rules out regions of minuscule background recombination rate in which sharp relative spikes in recombination still remain too small to be biologically interesting. The median is chosen here to be robust to the right skew of the distribution of recombination rates. Typically, for the human genome we use $\alpha_l = \alpha_r = 13$ kb, $\alpha_h = 2$ kb, and $k = 10$ based on experimental findings.

To apply our framework to the hotspot detection problem, we define the overall graphical model in Figure 2.2. The shaded nodes represent the observed variables. Denote w as a small window (typically < 25 kb) of the genome such that X_w is the population genetic data in that window, and X_{-w} is the rest. Similarly, let ρ_w and ρ_{-w} be the recombination map in the window and outside of the window, respectively. While ρ_w and ρ_{-w} have a weak dependence (dashed line) on X_{-w} and X_w respectively, this dependence decreases rapidly and is ignored for simplicity. More precisely, weak dependence means that $P(\rho_w, X_{-w}) \approx P(\rho_w)P(X_{-w})$ as shown in Equation 3.1 of [42] via a Taylor expansion argument. The intuition for this is that ρ tunes the correlation between neighboring sites so each site is effectively independent of recombination rates at distal sites.

Let q be the relative proportion of the sample possessing each mutation, and η be the population size function. Intuitively, η determines the rate at which the genealogies (can be thought of as binary trees) branch. q is a summary statistic of η which we observe that allows

us to fix the population size in an empirical Bayes style throughout training for simplicity using `SMC++`.

Let θ be the mutation rate and h be the indicator function for whether the window defines a hotspot. Conditioned on q, η is only weakly dependent on X_w.

We define our prior as follows. We sample the hotspot indicator variable $h \sim \text{Bernoulli}(0.5)$ and the local recombination maps $\rho_w \sim \hat{P}(\rho_w \mid h)$ from the released fine-scale recombination maps of HapMap [31]. The human mutation rate is fixed to that experimentally found in [53]. Since `SMC++` is robust to changes in any small fixed window, inferring $\hat{\eta}$ from X has minimal dependence on ρ_w.

To test for recombination hotspots:

1. Simulate a batch of h and ρ_w from the prior and X_w from `msprime` [49] given h and ρ_w.
2. Feed a batch of training examples into the network to learn $\mathbb{P}(h \mid X_w)$.
3. Repeat until convergence or for a fixed number of iterations.
4. At test time, slide along the genome to infer posteriors over h.

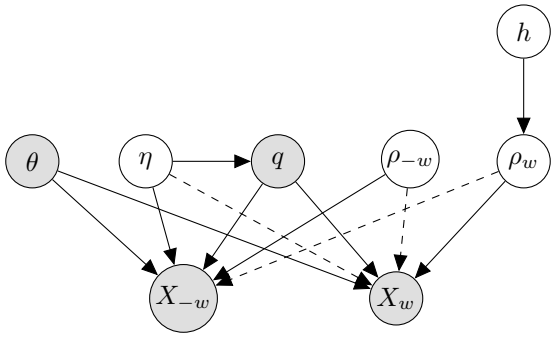

Figure 2.2: Graphical model of recombination hotspot inference: θ is the mutation rate, η the population size function, q the relative proportion of the sample possessing each mutation, ρ_{-w} the recombination rate function outside of the window, ρ_w the recombination rate function inside the window, h whether the window is a hotspot, X_{-w} the population genetic data outside of the window, and X_w the data inside the window. The dashed line signifies that, conditioned on q, η is weakly dependent on X_w for suitably small w, and ρ_{-w} and ρ_w are only weakly dependent on X_w and X_{-w}.

Evaluation of Exchangeable Neural Network

We compare the behavior of an explicitly exchangeable architecture to a nonexchangeable architecture that takes 2D convolutions with varying patch heights. The accuracy under

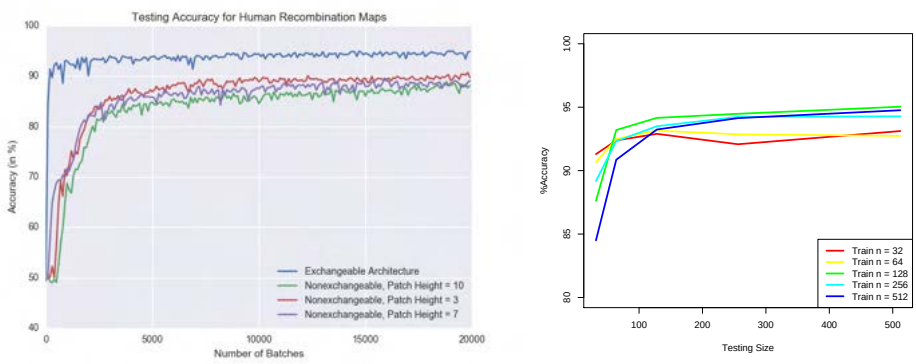

Figure 2.3: (Left)Accuracy comparison between exchangeable vs nonexchangeable architectures. (Right)Performance of changing the number of individuals at test time for varying training sample sizes.

human-like population genetic parameters with varying 2D patch heights is shown in the left panel of Figure 2.3. Since each training point is simulated on-the-fly, data augmentation is performed implicitly in the nonexchangeable version without having to explicitly permute the rows of each training point. As expected, directly encoding the permutation invariance leads to more efficient training and higher accuracy while also benefiting from a faster per-batch computation time. Furthermore, the slight accuracy decrease when increasing the patch height confirms the difficulty of learning permutation invariance as n grows. Another advantage of exchangeable architectures is the robustness to the number of individuals at test time. As shown in right panel of Figure 2.3, the accuracy remains above 90% during test time for sample sizes roughly 0.1–20$\times$ the train sample size.

Evaluation of Simulation-on-the-fly

Next, we analyze the effect of simulation-on-the-fly in comparison to the standard fixed training set. A fixed training set size of 10000 was used and run for 20000 training batches and a test set of size 5000. For a network using simulation-on-the-fly, 20000 training batches were run and evaluated on the same test set. In other words, we ran both the simulation on-the-fly and fixed training set for the same number of iterations with a batch size of 50, but the simulation-on-the-fly draws a fresh datapoint from the generative model upon each update so that no datapoint is used more than once. The weights were initialized with a fixed random seed in both settings with 20 replicates. Figure 2.4 (left) shows that the fixed training set setting has both a higher bias and higher variance than simulation-on-the-fly. The bias can be attributed to the estimation error of a fixed training set in which the empirical risk surface is not a good approximation of the population risk surface. The variance can be attributed to an increase in the number of poor quality local optima in the fixed training set case.

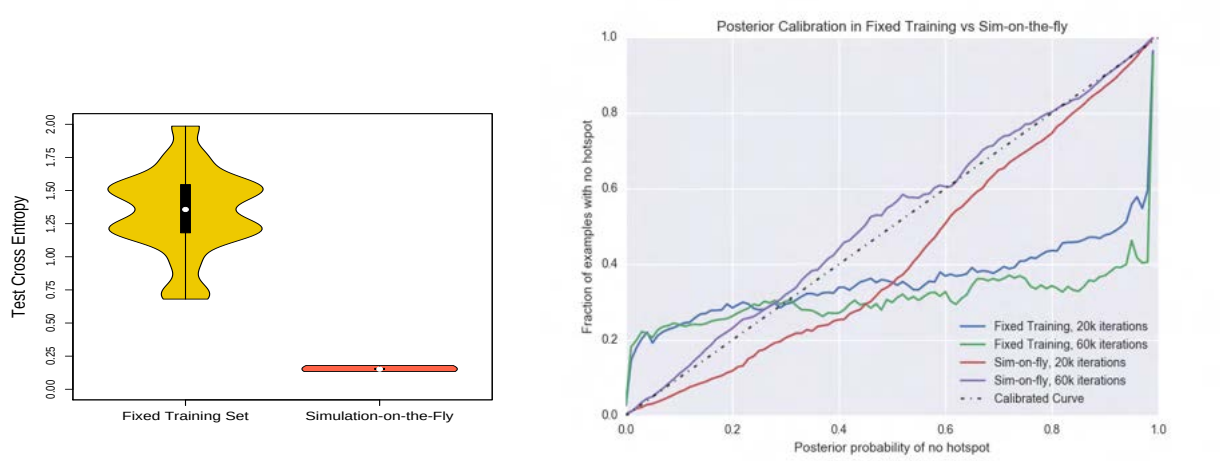

Figure 2.4: (Left)Comparison between the test cross entropy of a fixed training set of size 10000 and simulation-on-the-fly. (Right)Posterior calibration. The black dashed line is a perfectly calibrated curve. The red and purple lines are for simulation-on-the-fly after 20k and 60k iterations; the blue and green lines for a fixed training set of 10k points, for 20k and 60k iterations.

We next investigated posterior calibration. This gives us a measure for whether there is any bias in the uncertainty estimates output by the neural network. We evaluated the calibration of simulation-on-the-fly against using a fixed training set of 10000 datapoints. The calibration curves were generated by evaluating 25000 datapoints at test time and binning their posteriors, computing the fraction of true labels for each bin. A perfectly calibrated curve is the dashed black line shown in Figure 2.4 (right). In accordance with the theory in Section 2.3, the simulation-on-the-fly is much better calibrated with an increasing number of training examples leading to a more well calibrated function. On the other hand, the fixed training procedure is poorly calibrated.

Comparison to `LDhot`

We compared our method against `LDhot` in two settings: (i) sampling empirical recombination rates from the HapMap recombination map for CEU and YRI (i.e., Yoruba in Ibadan, Nigera) [31] to set the background recombination rate, and then using this background to simulate a flat recombination map with $10 - 100\times$ relative hotspot intensity, and (ii) sampling segments of the HapMap recombination map for CEU and YRI and classifying them as hotspot according to our definition, then simulating from the drawn variable map.

The ROC curves for both settings are shown in Figure 2.5. Under the bivariate empirical background prior regime where there is a flat background rate and flat hotspot, both methods performed quite well as shown on the left panel of Figure 2.5. We note that the slight

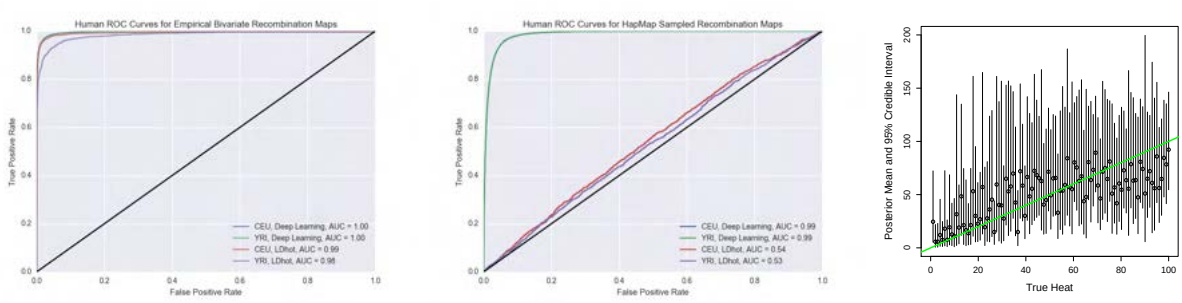

Figure 2.5: (Left) ROC curve in the CEU and YRI setting for the deep learning and `LDhot` method. The black line represents a random classifier. (Middle) Windows of the HapMap recombination map drawn based on whether they matched up with our hotspot definition. The blue and green line coincide almost exactly. (Right) The inferred posteriors for the continuous case. The circles represent the mean of the posterior and the bars represent the 95% credible interval. The green line shows when the true heat is equal to the inferred heat.

performance decrease for YRI when using `LDhot` is likely due to hyperparameters that require tuning for each population size. This bivariate setting is the precise likelihood ratio test for which `LDhot` tests. However, as flat background rates and hotspots are not realistic, we sample windows from the HapMap recombination map and label them according to a more suitable hotspot definition that ensures locality and rules out neglectable recombination spikes. The middle panel of Figure 2.5 uses the same hotspot definition in the training and test regimes, and is strongly favorable towards the deep learning method. Under a sensible definition of recombination hotspots and realistic recombination maps, our method still performs well while `LDhot` performs almost randomly. We believe that the true performance of `LDhot` is somewhere between the first and second settings, with performance dominated by the deep learning method. Importantly, this improvement is achieved without access to any problem-specific summary statistics.

Our approach reached 90% accuracy in fewer than 2000 iterations, taking approximately 0.5 hours on a 64 core machine with the computational bottleneck due to the `msprime` simulation [49]. For `LDhot`, the two-locus lookup table for variable population size using the `LDpop` fast approximation [46] took 9.5 hours on a 64 core machine (downsampling $n = 198$ from $N = 256$). The lookup table has a computational complexity of $O(n^3)$ while per-iteration training of the neural network scales as $O(n)$, allowing for much larger sample sizes. In addition, our method scales well to large local regions, being able to easily handle 800-SNP windows.

Recombination Hotspot Intensity Estimation: The Continuous Case

To demonstrate the flexibility of our method in the continuous parameter regime, we adapted our method to the problem of estimating the intensity (or heat) of a hotspot. The problem setup fixes the background recombination rate $R(w_l) = R(w_r) = 0.0005$ and seeks to estimate the relative hotspot recombination intensity k. The demography is set to that of CEU. The hotspot intensity k was simulated with a uniform distributed prior from 1 to 100.

For continuous parameters, arbitrary posteriors cannot be simply parameterized by a vector with dimension in the number of classes as was done in the discrete parameter setting. Instead, an approximate posterior distribution from a nice distribution family is used to get uncertainty estimates of our parameter of interest. This is achieved by leveraging our exchangeable network to output parameter estimates for the posterior distribution as done in [55]. For example, if we use a normal distribution as our approximate posterior, the network outputs estimates of the mean and precision. The corresponding loss function is the negative log-likelihood

$$-\log p(k|\mathbf{x}) = -\frac{\log \tau(\mathbf{x})}{2} + \frac{\tau(\mathbf{x})(k - \mu(\mathbf{x}))^2}{2} + \text{const}, \tag{2.3}$$

where μ and τ are the mean and the precision of the posterior, respectively. More flexible distribution families such as a Gaussian mixture model can be used for a better approximation to the true posterior.

We evaluate our method in terms of calibration and quality of the point estimates to check that our method yields valid uncertainty estimates. The right panel of Figure 2.5 shows the means and 95% credible intervals inferred by our method using log-normal as the approximate posterior distribution. As a measure of the calibration of the posteriors, the true intensity fell inside the 95% credible interval 97% of the time over a grid of 500 equally spaced points between $k = 1$ to 100. We measure the quality of the point estimates with the Spearman correlation between the 500 equally spaced points true heats and the estimated mean of the posteriors which yielded 0.697. This was improved by using a Gaussian mixture model with 10 components to 0.782. This illustrates that our method can be easily adapted to estimate the posterior distribution in the continuous regime.

2.6 Discussion

We have proposed the first likelihood-free inference method for exchangeable population genetic data that does not rely on handcrafted summary statistics. To achieve this, we designed a family of neural networks that learn an exchangeable representation of population genetic data, which is in turn mapped to the posterior distribution over the parameter of interest. Our simulation-on-the-fly training paradigm produced calibrated posterior estimates. State-of-the-art accuracy was demonstrated on the challenging problem of recombination hotspot testing.

The development and application of exchangeable neural networks to fully harness raw sequence data addresses an important challenge in applying machine learning to population genomics. The standard practice to reduce data to ad hoc summary statistics, which are then later plugged into a standard machine learning pipelines, is well recognized as a major shortcoming. Within the population genetic community, our method proves to be a major advance in likelihood-free inference in situations where ABC is too inaccurate. Several works have applied ABC to different contexts, and each one requires devising a new set of summary statistics. Our method can be extended in a black-box manner to these situations, which include inference on point clouds and quantifying evolutionary events.

2.7 Proofs

Proof of Proposition 1 By the Universal Approximation Theorem and the interpretation of simulation-on-the-fly as minimizing the expected KL divergence between the population risk and the neural network, the training procedure minimizes the objective function for any $\mathbf{x}$, $\epsilon > 0$, $\delta > 0$, we can pick a $H > H_0$, and $N > N_0$ such that,

$$\min_{\mathbf{w}} \mathbb{E}_{\mathbf{x}}\Big[KL\Big(\mathbb{P}(\theta \mid \mathbf{x}) \;\Big\|\; \mathbb{P}_{DL}^{(N)}(\theta \mid \mathbf{x}; \mathbf{w}, H)\Big)\Big] < \epsilon.$$

Let $\mathbf{w}^*$ be a minimizer of the above expectation. By Markov's inequality, we get for every $\mathbf{x}$ and $\delta > 0$ such that for all $H > H_0$ and $N > N_0$

$$KL\Big(\mathbb{P}(\theta \mid \mathbf{x}) \;\Big\|\; \mathbb{P}_{DL}^{(N)}(\theta \mid \mathbf{x}; \mathbf{w}^*, H)\Big) < \delta$$

with probability at least $1 - \frac{\epsilon}{\delta}$. □

Proof of Corollary 1 As above, for any $\mathbf{x}$, $\epsilon > 0$, $\delta > 0$, there exists a $H > H_0$, and $N > N_0$ such that

$$\min_{\mathbf{w}} \mathbb{E}_{\mathbf{x}}\Big[KL\Big(\mathbb{P}(\theta \mid \mathbf{x}) \;\Big\|\; \mathbb{P}_{DL}^{(N)}(\theta \mid \mathbf{x}; \mathbf{w}, H)\Big)\Big] < \epsilon.$$

Furthermore, for all $\mathbf{x}$, the KL is bounded at the minimizer since $\mathbb{P}(\mathbf{x}) > 0$ for all $\mathbf{x}$ resulting in the following bound

$$KL\Big(\mathbb{P}(\theta \mid \mathbf{x}) \;\Big\|\; \mathbb{P}_{DL}^{(N)}(\theta \mid \mathbf{x}; \mathbf{w}^*, H)\Big) < \max_{\mathbf{x}} \frac{\epsilon}{\mathbb{P}(\mathbf{x})}$$

independent of $\mathbf{x}$. Thus, the training procedure results in a function mapping that uniformly converges to the posterior $\mathbb{P}(\theta \mid \mathbf{x})$. □

Chapter 3

Archaic Admixture Detection

This is joint work with Yun S. Song. This was first presented at the *NeurIPS Machine Learning in Computational Biology* [20].

3.1 Introduction

Reconstructing the evolutionary history of how humans adapted to changing environments and diets has received significant momentum in the past few years with the explosion of ancient DNA samples. Such biological understanding could provide medical insights into today's ever-changing human lifestyle and environment. Admixture is one such evolutionary mechanism for adaptation by inducing genetic variation in human populations. For example, recent evidence has shown that admixture between Neanderthals and Eurasian populations are responsible for diversity in immune genes [24].

A key challenge is improving the computational tools utilized for identifying portions of our genomic adaptations attributed to interbreeding between modern humans and archaic hominids. Recent studies [75, 74] have developed computational tools for analyzing the interbreeding between modern Eurasian humans and archaic hominids (Neanderthals and Denisovans). Their state-of-the-art utilizes a conditional random field (CRF) dependent on access to archaic reference genomes to detect segments of archaic DNA. Reference-dependent methods heavily rely on the archaic reference, known to be extremely noisy and contaminated by modern human DNA. Furthermore, these methods are unable to identify admixture from a population where DNA samples are unavailable (often referred to as a ghost population). In a separate line of work, a reference-free method S^* [65, 92] has been developed to test for the existence of admixed archaic DNA. Using S^*, the authors conjecture the existence of admixture with a ghost population in the African genome [36] though remain unable to provide sufficient statistical evidence to confidently verify the claim. Both of these methods are unable to directly combine information between individuals and neighboring SNPs at multiple scales; instead, the methods reduce the data down to more manageable summary statistics.

Our method tackles both of these problems and significantly improves on the state-of-the-art in both classification and segmentation. Our method utilizes a deep learning-based Bayesian inference framework to learn a mapping between the raw genetic data and the latent variable of interest (archaic DNA or not). The exchangeability between DNA samples within a population requires the development of a permutation-equivariant deep neural network in order to fully leverage the information across all samples. In addition, our Bayesian inference framework can elegantly incorporate any known inferred parameters such as archaic reference genomes and recombination maps while also showing robustness to uncertainty over unknown population genetic parameters.

In this chapter, we propose a more general permutation equivariant layer, demonstrate its efficacy in the population genetics setting, and devise the first usage of permutation-equivariant networks to the classification and segmentation tasks. We further show that our segmentation method is the first reference-free method to identify admixed segments with sufficient accuracy.

3.2 Structured Coalescent with Recombination

The coalescent [52, 51] is a useful generative stochastic process in evolutionary biology for describing the genealogical history of a random sample of chromosomes. The structured coalescent with recombination incorporates evolutionary processes important for understanding and characterizing the signatures of ghost admixture including coalescence, mutation, recombination, and migration.

At a high level, the structured coalescent (see the left panel of Fig. 3.1) is the limiting process by which ancestral trees trace the history of relatedness among present-day individuals. In the absence of recombination, the coalescent process can be thought of as a generative process over binary trees. Coalescent events describe the merging of children nodes (times at which two individuals shared an ancestor) occurring at a rate parameterized by the size of the subpopulation. Migration describes the movement of individuals between subpopulations. Once the tree of evolutionary history is constructed, mutations are dropped along the tree branches. All events are distributed as a Poisson point process. In the presence of recombination, there is a tree marginally at each base pair of the genome with the recombination rate tuning the degree to which neighboring trees are correlated.

The coalescent model informs us of the types of genetic signatures that we expect from archaic admixture. Long contiguous blocks of DNA with mutations private to the ghost population genome due to accumulated mutations in the ghost population is one signature of archaic admixture. The archaic reference genome would inform us which alleles are present in the archaic population to boost accuracy.

Typical datasets in population genetics are sets of DNA samples from present-day. Most inference tasks are then to recover the evolutionary parameters of the coalescent given data sampled marginally at present-day. Inference is often intractable without simplifying as-

sumptions as the likelihood is often analytically and computationally intractable since the state space of possible trees is super-exponential in n.

While difficult to compute likelihoods, the coalescent is simple to draw samples from $X \sim p(X \mid \theta)$ for population genetic parameters θ. Software packages such as `msprime` [49] can efficiently simulate data given population genetic parameters allowing for computationally feasible likelihood-free inference methods.

For more details on coalescent theory, refer to [26].

3.3 Method

Permutation Equivariant Layer

With the goal of designing a neural network that can take in genotype data, we focus on designing a neural network architecture that respects the permutation-equivariant symmetries inherent in the genotype data. To do this, we need to design a neural network layer where permutations of the input yield permutations of the output.

Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)^T \in \mathbb{R}^{n \times d}$ be a matrix (which in our example represents a single genotype array where the rows represent the individuals and the columns represent the SNPs) and S_n denote the symmetric group. We wish to construct a neural network layer Φ that is *equivariant* with respect to all row-permutations, that is, for every $g \in S_n$,

$$g(\Phi(\mathbf{X})) = \Phi(g(\mathbf{X})).$$

We use parameter sharing to encode permutation equivariance for computational efficiency and to prevent a combinatorial explosion of parameters. Our proposed layer defines the ith hidden unit as

$$\Phi(\mathbf{X}; \mathbf{w}_{\text{self}}, \mathbf{w}_{\text{other}})_i = \sigma\big(\mathbf{x}_i \cdot \mathbf{w}_{\text{self}} + f(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots; \mathbf{w}_{\text{other}})\big),$$

where σ is a pointwise nonlinearity and $f : \mathbb{R}^{n-1} \to \mathbb{R}$ is any symmetric function parameterized by $\mathbf{w}_{\text{other}}$ such as the max, sum, sort, or higher-order moments. Such a neural network layer can replace multiple neural network layers since the composition of two such layers remains permutation equivariant, i.e. $\Phi_1(\Phi_2(g(x))) = \Phi_1(g(\Phi_2(x))) = g(\Phi_1(\Phi_2(x)))$. This is a generalization of the permutation equivariant layers proposed by [71], [70], and [98].

Structured Permutation Equivariant Layer

Often for archaic admixture analyses incorporating multiple populations of data can significantly improve performance either as an outgroup or the archaic population of interest. This requires the neural network layer to obey permutation-equivariance within a population but not between populations. Let there instead be multiple populations such that the genotype

array for population j is defined as $\mathbf{X}^{(j)} = (\mathbf{x}_1^{(j)}, \mathbf{x}_2^{(j)}, \ldots, \mathbf{x}_{n_j}^{(j)})^T \in \mathbb{R}^{n_j \times d}$. Now we wish to construct a neural network layer Φ that for every $g_j \in S_{n_j}$ satisfies

$$g_j \cdot \Phi(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(J)})_j = \Phi(g_1 \cdot \mathbf{X}^{(1)}, \ldots, g_j \cdot \mathbf{X}^{(j)}, \ldots, g_J \cdot \mathbf{X}^{(J)})_j.$$

Once again we appeal to parameter sharing to encode this symmetry. Our proposed layer defines for hidden unit i in population j

$$\Phi(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(J)}; \mathbf{w}_{\text{self}}, \mathbf{w}_{\text{other}}, \mathbf{w}_{\text{across}})_{ij} = \sigma\left(\mathbf{x}_i^{(j)} \cdot \mathbf{w}_{\text{self},j} + f(\mathbf{x}_{-i}^{(j)}; \mathbf{w}_{\text{other},j}) + \sum_k h_k(\mathbf{X}^{(k)}; \mathbf{w}_{\text{across},j,k})\right)$$

where parameters $\mathbf{w}_{\text{self}}, \mathbf{w}_{\text{other}}, \mathbf{w}_{\text{across}}$ are of size $\mathbb{R}^J, \mathbb{R}^J,$ and $\mathbb{R}^{J \times (J-1)}$, respectively and we define $\mathbf{x}_{-i}^{(j)}$ as all units in population j excluding unit i. In addition, f and h_k are symmetric functions as in the prequel.

Likelihood-Free Inference Framework

We modify the likelihood-free inference framework developed in [21] by incorporating an additional latent variable.

Framework for inference of parameters $\theta = (\alpha, t_{ad}, t_{anc})$ (admixture proportion, admixture time, and ancestral time, respectively as demonstrated in Fig. 3.1), latent segmentation parameters $\mathbf{Z} \in \{0,1\}^{n \times L}$, and population genetic data $\mathbf{X} \in \{0,1\}^{n \times L}$:

- Simulate training data on-the-fly from the prior $\theta_i \sim p(\theta)$ and $\mathbf{X}_i, \mathbf{Z}_i \sim p(\mathbf{X}, \mathbf{Z} \mid \theta_i)$ via `msprime`.
- Train a permutation equivariant network to learn the posterior $p(\mathbf{Z}_i \mid \mathbf{X}_i)$
- Estimate parameters of interest $p(\theta \mid \mathbf{X}_i) = \int_Z p(\theta \mid \mathbf{Z}_i, \mathbf{X}_i) p(\mathbf{Z}_i \mid \mathbf{X}_i) d\mathbf{Z}_i$.

In the case of ghost admixture, $p(\theta \mid \mathbf{Z}_i, \mathbf{X}_i) = p(\theta \mid \mathbf{Z}_i)$. Some parameters such as recombination rate and population size functions are only weakly dependent on $\mathbf{Z}_i$, so we use an empirical Bayes approach by inferring the recombination map ρ and population size function $N(t)$ separately and fixing the maximum likelihood estimate in the prior $p(\theta, \hat{\rho}, \hat{N})$.

Due to the imbalanced number of examples of introgressed segments in the data for both classification and segmentation, we employ an importance sampling-type weighting scheme of the loss function to balance the classes. For admixture proportion α, we upweight the loss for ghost DNA by $\frac{1-\alpha}{\alpha}$ such that the classes are balanced.

Network Architecture

Our network architecture for classification (see Figure 3.1) uses the architecture of a standard convolutional neural network with all of the convolutions replaced by our convolutional

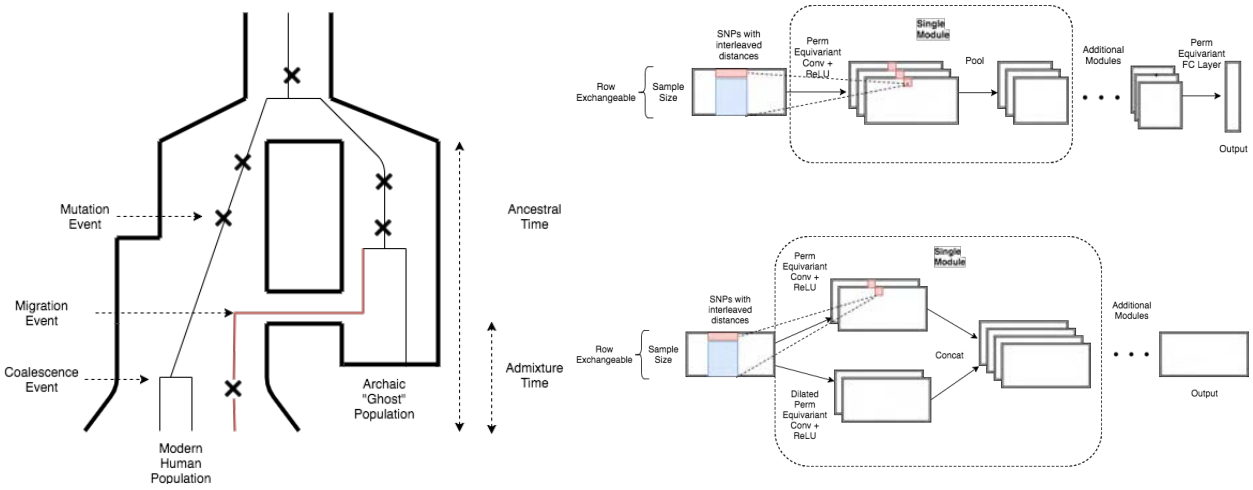

Figure 3.1: (Left) An example of the coalescent with ghost admixture. (Top) The permutation-equivariant network for classification. (Bottom) The permutation-equivariant network for segmentation.

permutation equivariant layer followed by pooling layers. The final layer is a fully connected permutation equivariant layer.

Our network architecture for segmentation borrows ideas from fully convolutional networks [60] and concatenates together convolutional permutation equivariant layer outputs with their dilated counterparts to capture local and global information.

Estimation of Archaic Admixture Parameters

Additional parameters related to archaic admixture are typically of interest to be jointly inferred including ancestral time (time at which the sample split from the modern population) and admixture time (time at which the sample was admixed). An unbiased estimator for the admixture proportion α is simply the average posterior of the segmentation $\hat{\alpha} = \frac{1}{nL}\sum_{i,j} p(y_{ij} = 1 \mid \mathbf{X})$ where y_{ij} is the label for the ith individual and position j with class label 1 indicating presence of admixture. However, due to our importance sampling-style class balancing scheme the neural network posteriors need to be re-calibrated. We recalibrate the posterior by computing

$$p(y_{ij} = 1 \mid \mathbf{X}) = \frac{\alpha q(y_{ij} = 1 \mid \mathbf{X})}{\alpha q(y_{ij} = 1 \mid \mathbf{X}) + (1 - \alpha) q(y_{ij} = 0 \mid \mathbf{X})}$$

where q is the uncalibrated posterior by the neural network. Note that in the case where α is not known, we can integrate over our prior $p(\alpha)$.

The admixture time t_{ad} can be estimated from Z_i via $L_{tracts} \sim \text{Exp}(\frac{\rho}{2} t_{anc})$ where L_{tracts} and $\rho/2$ is the length of contiguous ghost introgressed segments and the recombination rate, respectively. The ancestral time t_{anc} can be inferred from Tajima's estimator [87].

Incorporating Additional Genetic Information

In many admixture settings, the inclusion of an outgroup reference genome which is known to not have admixed with the ghost population can boost accuracy of classification and segmentation. The outgroup can be easily incorporated as an additional input by using the structured permutation equivariant neural network. A similar approach can be used with a known archaic reference such as for Neanderthals or Denisovans.

Statistical power for detecting signatures of archaic admixture is heavily dependent on linkage disequilibrium (determined by the recombination rate) to allow for the sharing of statistical strength between neighboring sites. Variable recombination maps can be similarly incorporated as a separate channel of the input.

3.4 Experiments

Our experiments are based on two demographic models to most accurately compare to the state-of-the-art. In the reference-based and reference-free settings, we use the demographic models proposed in [75] and [92], respectively. The first demographic model was used to infer archaic admixture between European populations and Neanderthals, so parameters are set to $\alpha = 0.03, t_{anc} = 13000$ generations, and $t_{ad} = 1900$ generations. The second demographic model was used to posit ghost admixture in African populations with parameters $\alpha = 0.03, t_{anc} = 28000$ generations, and $t_{ad} = 1400$ generations. For simplicity, the per-generation mutation and recombination rate were set to realistic human rates of $\mu = 1.5 \times 10^{-8}$ and $r = 1.2 \times 10^{-8}$, respectively. However, as noted in the prequel, simulations using fine-scale recombination maps or time-varying mutation rates can be used and in principle even jointly estimated.

The neural network architecture contains 5 hidden layers with standard 1D convolutions with patch size of 5 (with max pooling between layers in the classification setting) and 64 filters in each hidden layer. In the third and fourth layers, dilated convolutions were performed and concatenated with dilations of size 4 to better capture multiscale information. Training was performed with batch sizes of 50 run for 20000 training iterations.

We performed experiments in three setups:

- *Segmentation with Reference*: Segmentation of each SNP and individual in the presence of a reference in comparison to the archaic reference-dependent CRF of [75].
- *Reference-Free Classification*: Classification of each individual in a population as possessing archaic hominid DNA in comparison to the S^* statistic [92].

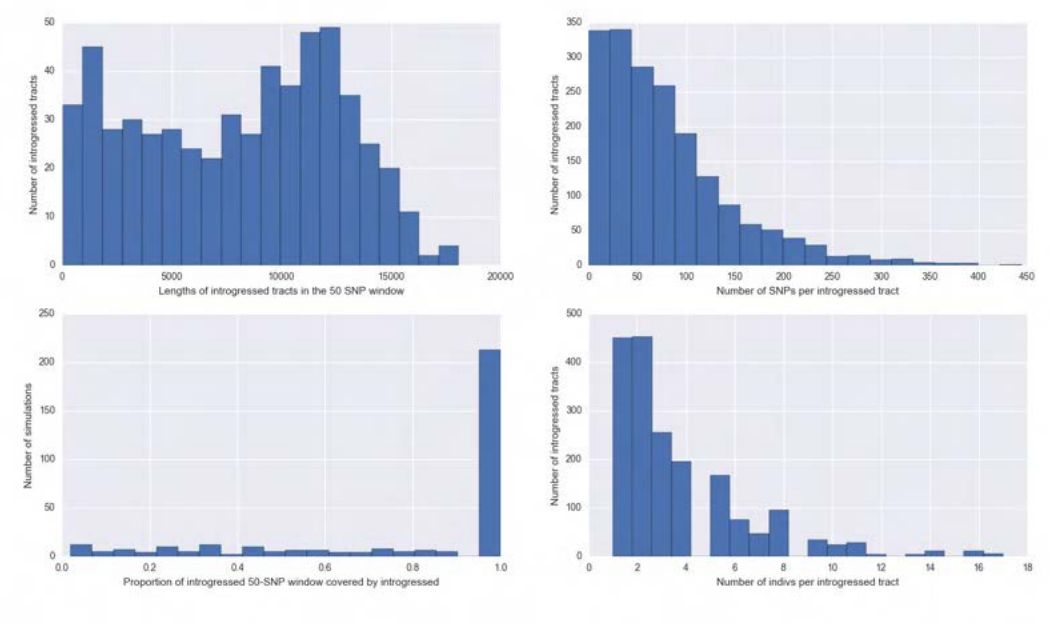

Figure 3.2: Simulation analysis of the model. Top left: The histogram of physical distance of introgression tracts. Top right: The histogram of number of SNPs within a introgression tract. Bottom left: Conditioned on an individual having an introgression tract in a 50 SNP window (default width of the sliding genotype array window), the proportion of the window that is introgressed. Bottom right: Number of individuals with an introgressed tract in a 50 SNP window conditioned on at least one individual being admixed.

- *Reference-Free Segmentation*: Reference-free segmentation of each SNP and individual.

We analyze the distribution of introgression tracts under the simulation model in Fig. 3.2. The number of individuals who observe introgression at the same position drops off fairly rapidly which agrees with the demographic model as shown in the top right of the figure. Furthermore, most individuals who see any introgression within a 50 SNP window see a large proportion of its SNPs as introgressed as shown in the bottom left. This indicates that there is a high-level of class imbalance in this experimental setup.

Next, we experiment across a host of symmetric functions in Fig. 3.3 to derive a better understanding for the properties of our structured permutation-equivariant network. Reassuringly, we observe overall the performance does not depend heavily on the choice of symmetric function and symmetric functions that carry more information (sort-based functions rather than single statistics such as mean and max) tend to slightly outperform others at test time.

We analyze the reference-based segmentation of our procedure against that of [75]. We

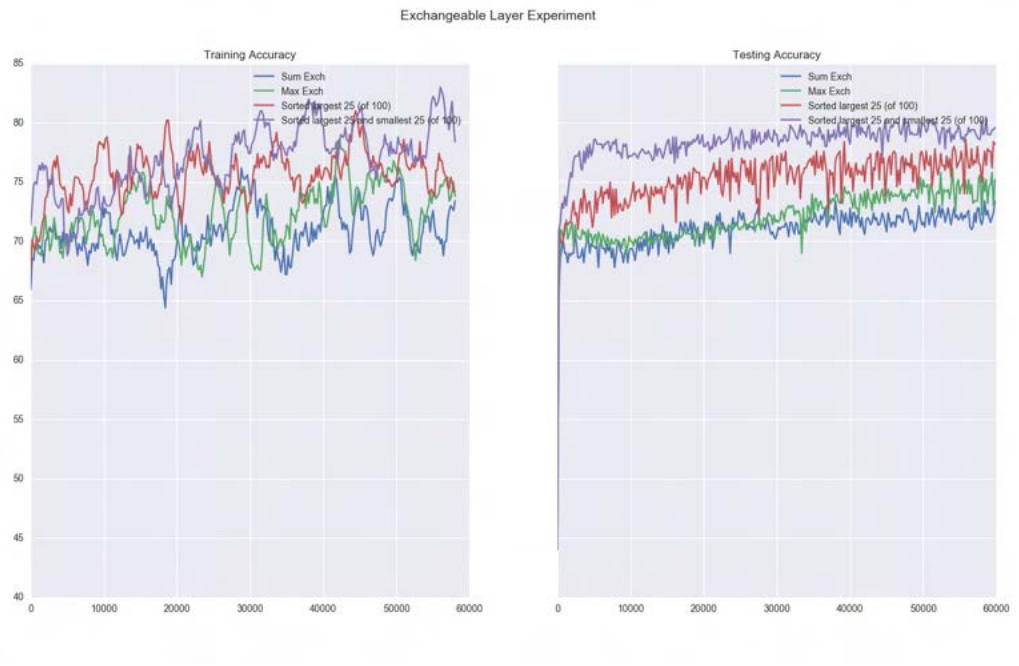

Figure 3.3: Comparison between a variety of symmetric functions for the permutation equivariant network. The four functions used are top-25 (red), top-25 concatenated with bottom-25 (purple), max (green), and sum (blue).

note that we had difficulty replicating the results in the paper under the experimental conditions described in the paper with the exact reasoning unknown after communication with the authors. Instead, we took the direct precicion-recall curve produced in the paper and compared it against our method in Fig. 3.4. Note that as was shown in the simulation analysis that precision-recall is a more appropriate metric for performance than ROC curves due to class imbalance. We outperform the CRF method with fewer individuals in the outgroup across all recall levels. This may be due to the CRF's very simplistic hand-crafted feature functions which are unable to properly aggregate local and global spatial information. Incorporating multi-scale spatial information allows us to share statistical strength between correlated SNPs to make comparable predictions despite the absence of a reference. Furthermore, our method uses fewer individuals (20 vs 100 in the CRF method) in the outgroup to reach superior performance. The qualitative jaggedness of the precision-recall can be attributed to the edge effects of our sliding window procedure which can be smoothed out post-hoc via an averaging method.

We compared against the only other reference-free method, S^*, for classification as shown in Fig. 3.5. Our method significantly outperforms S^* which we attributed to the relative brittleness of S^* to account for slight changes in demography. The structured permutation-equivariant layer significantly outperforms that of just classifying each haplotype one at a

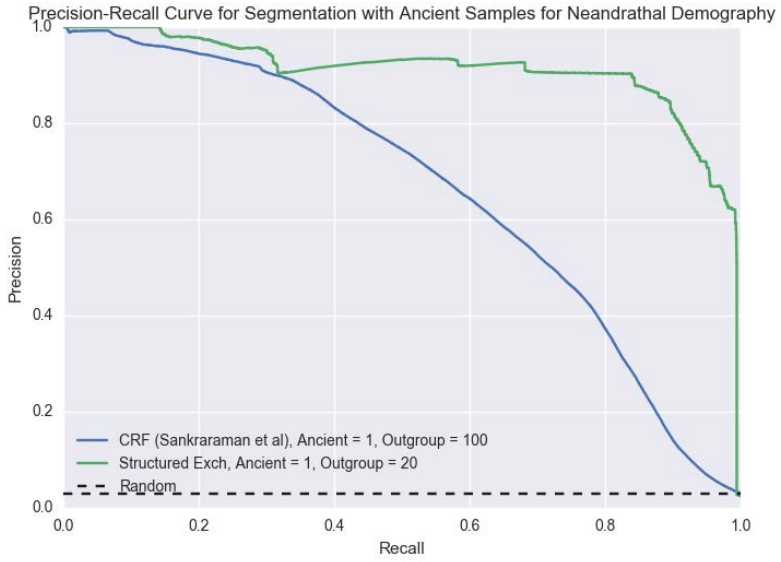

Figure 3.4: Comparison of our method (green) against that of the CRF method (blue). The dashed line is the precision recall of a random classifier. This method was evaluated on a single ancient genome and 20 individuals in the outgroup for our method and 100 individuals in the outgroup for the CRF.

time which demonstrates the utility of our structured permutation-equivariant layer and the information-sharing between haplotypes.

While no other methods have been developed to segment tracts of ghost archaic DNA (DNA without an archaic hominid reference genome), several methods have been utilized in the presence of a known archaic hominid population such as Neandrathals and Denisovans [77, 67, 75]. Since neither [77] nor [67] validates their method on simulated data and [67] admits that their method has less statistical power for detecting archaic admixture in comparison to [75], we do not compare our method against any baselines. In Fig. 3.6, we demonstrate that our method in the absence of an archaic hominid reference achieves rather strong precision-recall. In addition, our method achieves significant improvements when performing post-hoc smoothing similar to the procedure performed on the CRF to account for longer-range interactions.

To test the ability of our method to estimate the admixture proportion α, we simulated 100 haplotypes over 100 SNP windows from a realistic uniform prior $\alpha \sim U(0.01, 0.1)$. The mean squared error $\frac{1}{n}\sum_{i=1}^{n}(\alpha_i - \hat{\alpha}_i)^2$ of our estimator was 0.0016 and our accuracy remained robust. The error is expected to be even smaller when averaging estimates over genome-scale data.

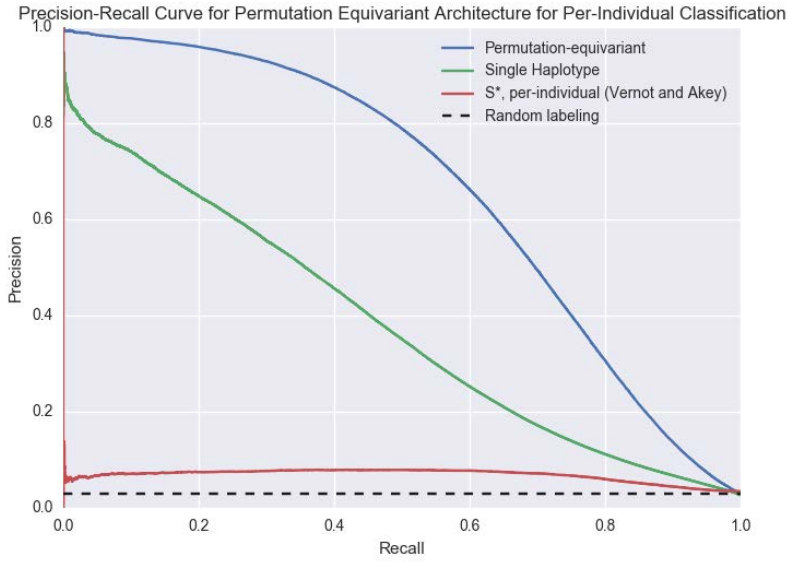

Figure 3.5: Comparison on the reference-free classification task of our permutation-equivariant procedure (blue) and a single haplotype procedure which does not jointly classify individuals in a population (green) and S^* (red).

3.5 Discussion

We have proposed a novel structured permutation-equivariant neural network which accounts for the symmetries and structure of archaic admixture segmentation and classification. This structure accounts for multiple populations which do not observe permutation-equivariance between members of groups but observe this property for members within groups. Applying our method inside a statistical inference framework allows us to outperform other methods with and without an archaic reference genome.

The development of this procedure enables the population genomics community to address fundamental questions around the evolutionary history of modern populations. This approach can extend further to other questions where per-SNP resolution is desirable such as standard admixture. Additional work on the inference procedure regarding the robustness of the procedure to model misspecification in the demography or other fundamental population genetic parameters is essential to the exploration of this methodology on real-world data.

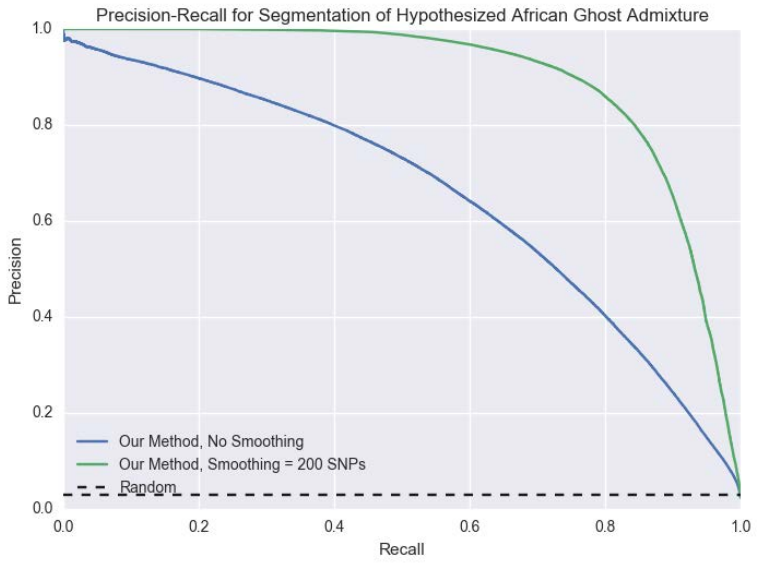

Figure 3.6: Precision-recall curves on the reference-free segmentation task of our permutation-equivariant procedure with (green) and without(blue) post-hoc smoothing of up to 200 SNPs.

Chapter 4

Parallelizing Contextual Linear Bandits for Protein Engineering

This is joint work with Aldo Pacchiano, Nilesh Tripuraneni, Peter Bartlett, Yun S. Song, and Michael I. Jordan. This work will soon be published as a preprint.

4.1 Introduction

Traditional approaches to adaptive design and search often require time and resource-intensive modeling. However, as the cost of experimentation has steadily dropped, the amount of *real time* needed for search has become the primary constraint. For example, within the field of biology, advances in synthesis and high-throughput sequencing have shifted the the focus from top-down mechanistic modeling towards iterative, data-driven search algorithms [10]. In protein engineering, the Nobel prize-winning approach of directed evolution [8]–which mimics evolution via iterative rounds of measurement of modified sequences–has already been successful in improving therapeutic antibodies [37] and changing substrate specificity [78]. Within the realm of information technology, learning in a variety of real-world interactive tasks is also well-suited to adaptive, data-driven search: such as a recommender system which generates movie suggestions for users on the basis of binary feedback for previously recommended movies [15]. An important property of both of these examples, is that learning must be done via bandit feedback. Given the state of a system (i.e. a particular protein sequence or user) the corresponding value or feedback is only observed for that particular state.

Under these models of limited feedback, utilizing adaptive, sequential approaches to *intelligently* query the design space is critical. In recent years, bandit methods and Bayesian optimization have been used to great effect in a variety of applications to navigate the exploration-exploitation trade-off in the design space. The ultimate goal of such approaches is to minimize the total amount of *real time* (equivalent to the number of sequential measurements) needed to learn the corresponding application domain.

However, simply designing better methods that are more efficient through time neglects an orthogonal axis through which progress is possible: parallelism at a fixed time. In many applications of interest, it is often feasible to perform parallel measurements (or batch queries) simultaneously. In protein engineering, batch queries can be as large as 10^6 sequences while each query takes months to measure [82]. Similarly, large-scale recommender systems can often make multiple, concurrent interactions with different users [3]. Parallelism provides a mechanism to utilize further hardware for progress, while not increasing the amount of *real time* needed to learn. The drawback of making batch queries at a fixed time is that queries within a batch will necessarily be less informed then sequential queries–since their choice cannot benefit from the results of other experiments within the same batch.

Here we attempt to characterize the utility of parallelism in the simple setting of (contextual) linear bandits. In particular, we ask: can parallelism provide the same benefit as sequentiality through time in this class of adaptive decision-making problems? Perhaps surprisingly, and mirroring earlier work in distinct settings, we show the answer is often yes. We consider the setting where P distinct processors/machines perform simultaneous queries in batches of size P, $\{\mathbf{x}_{t,p}\}_{p=1}^{P}$, over T distinct rounds, to an approximately linear, noisy reward oracle $r_{t,p} \approx \mathbf{x}_{t,p}^\top \boldsymbol{\theta}^\star + \epsilon_{t,p}$. Importantly, the reward feedback for all P processors in a given round, is only observed after the entire batch of all P actions has been selected. In this setting, we ask what the price of parallelism is with respect to a single, perfectly sequential agent querying the same reward oracle over TP rounds. Our results show that up to a burn-in term independent of time, the worst-case regret of parallel, contextual linear bandit algorithms can nearly match their perfectly sequential counterpart. Informally, under the standard bandit setting where the signal-to-noise ratio is taken to be $\Theta(1)$, and d is the dimension of the action-space, our results show that (parallel) variants of optimistic linear bandit algorithms can achieve aggregate regret,

$$\tilde{O}(P \cdot \kappa + d\sqrt{TP}),$$

where κ is a term capturing the geometric complexity of the sequence of contexts. For *arbitrary* sequences of contexts we have that $\kappa = \tilde{O}(d)$, but for sequences of contexts with additional geometric structure we provide sharper instance-dependent bounds for κ. The second term in the former regret matches that of a purely sequential agent playing for TP rounds, $\tilde{O}(d\sqrt{TP})$. The first burn-in term represents the price of parallelism which is subleading whenever $TI pre \gtrsim \tilde{\Omega}(P\kappa^2/d^2)$. However, we note for many applications it may be the case that $P \gg T$ in the regimes of interest–so understanding the scaling of κ is an important question. Our work is motivated by the class of design problems in which *statistical sample complexity* is the primary object of interest. In between batch queries at different time steps, the cost of inter-processor communication and computation is often negligible. In this setting we make the following contributions:

- We introduce a family of parallelized linear bandit algorithms applicable in the general setting of contextual bandits (which encompasses potentially changing and infinite

action sets), building on classic, optimistic algorithms such as Linear UCB and Linear Thompson sampling.

- We present a unified treatment of these algorithms which allows a fine-grained understanding of the interaction of the scales of the underlying parameters, model misspecification, parallelism, and context set geometry. In particular, our analysis highlights the scale of the burn-in term $\propto P$. Although asymptotically negligible as $T \to \infty$, this term can be significant in the regime where we have $TP \gg \text{poly}(d)$ but $P \gg T$–which is realistic for many real applications.

- We provide regret lower bounds for the misspecified, linear bandit problem in the parallel setting, confirming our algorithms are optimal up to logarithmic factors as the time horizon tends to $T \to \infty$.

- We present a comprehensive experimental evaluation of our parallel algorithms on a suite of synthetic and real problems. Through our theory and experiments, we show the importance of explicitly introducing diversity into the action selections of parallel bandit algorithms since it often leads to practical performance gains.

Related Work

Parallel sequential decision-making problems often occur in high-throughput experimental design of synthetic biology applications. For protein engineering, [72] employs batch-mode Gaussian Process Upper Confidence Bound(GP-UCB) while [14] uses the batched Bayesian optimization via parallelized Thompson sampling proposed in [47]. [7] proposes a portfolio optimization method layered atop an ensemble of optimizers. Finally, [83] utilizes a heuristic, batched greedy evolutionary search algorithm for design. Similarly, it is well-known that contextual linear bandit algorithms are often used in dynamic settings on large, multi-user platforms for problems such as ad placement, web search, and recommendation [85, 15]. However, the empirical study of such domains (including our own) is limited due to the paucity of publicly available nonstationary data sources.

Several investigations of the utility of parallelism in sequential decision-making problems take place in the framework of Gaussian process (GP)-based Bayesian optimization. [23] shows in a Bayesian framework, a lazy GP-bandit algorithm initialized with uncertainty sampling, can achieve a parallel regret nearly matching the corresponding sequential regret up to a "burn-in" term independent of time. Later, [48] showed diversity induced by determinantal point processes (DPP) can induce additional, useful exploration in batched/parallel Bayesian optimization in a comparable setting. [47] establishes regret bounds for Bayesian optimization parallelized via Thompson sampling algorithm obtaining qualitatively similar theoretical results to [48, 23].

A related line of work studies bandit learning under delayed reward feedback. [22] provides an early, empirical investigation of Thompson sampling under fixed, delayed feedback.

Several theoretical works in [45, 90, 91, 100] prove regret bounds for bandit algorithms under known and unknown *stochastic* delayed feedback models. Here the reward information is delayed by a random time interval from the time the action is proposed, and is distinct from our setting from the batch queries result in a fixed time delay. In the setting of finite-armed contextual bandits [25] Delayed Policy Elimination algorithm satisfies a regret bound of the form $\tilde{\mathcal{O}}(\sqrt{m}(\sqrt{T}+\tau))$ where m denotes the number of actions, T the problem horizon and τ the delay. However, these guarantees require i.i.d. stochastically generated contexts and require access to a cost-sensitive classification oracle for their elimination-based protocol.

A more closely related line of work studies distributed bandit learning under various models of limited communication between agents (which can make decisions in parallel). [39, 86] study the regret of distributed arm-selection algorithms under restricted models of communication between parallel agents in the setting of multi-armed bandits. [54] and [96] provide distributed confidence-ball algorithms for linear bandits in peer-2-peer networks with a focus towards limiting communication complexity. However, the algorithms in [54] and [96] both require *intra-round communication* of rewards, that is incompatible with the batch setting that we study in this work. Moreover these works are mostly concerned with the regime $T \gg P$, reducing their practical relevance.

Our work provides a class of algorithms and regret analysis for the general linear contextual bandit problem in the batch setting. Moreover, our goal is to design theoretically sound and practically useful algorithms for parallel linear bandits. Accordingly, we construct several algorithms we analyze both theoretically and empirically.

Preliminaries

Set-up and Notation Throughout, we will use bold lower-case letters (e.g., $\mathbf{x}$) to refer to vectors and bold upper-case letters to refer to matrices (e.g., $\mathbf{X}$). The norm $\|\cdot\|$ appearing on a vector or matrix refers to its ℓ_2 norm or spectral norm respectively. A matrix-subscripted norm $\|\mathbf{x}\|_{\mathbf{\Sigma}} = \sqrt{\mathbf{x}^\top \mathbf{\Sigma} \mathbf{x}}$ for positive semi-definite matrix $\mathbf{\Sigma}$. $\langle \mathbf{x}, \mathbf{y} \rangle$ is the Euclidean inner product. Generically, we will use "hatted" vectors and matrices (e.g., $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathbf{B}}$) to refer to (random) estimators of their underlying population quantities. We also use the bracketed notation $[n] = \{1, \ldots, n\}$. We will use $\gtrsim$, $\lesssim$, and $\asymp$ to denote greater than, less than, and equal to up to a universal constant and use $\tilde{O}$ to denote an expression that hides polylogarithmic factors in all problem parameters. Our use of O, Ω, and Θ is otherwise standard.

Formally, we consider the (parallel) contextual linear bandit setting where at each round t, the p-th bandit learner receives a context $\mathcal{X}_{t,p} \subset \mathbb{R}^d$ and a master algorithm $\mathcal{A}$ commands each learner to select an action $\mathbf{x}_{t,p} \in \mathcal{X}_{t,p}$ on the basis of all the past observations. Given an (unobserved) function $f(\cdot)$ each learner simultaneously receives a noisily generated reward:

$$r_{t,p} = f(\mathbf{x}_{t,p}) + \xi_{t,p}, \tag{4.1}$$

where $\xi_{t,p}$ is an i.i.d. noise process and $f(\mathbf{x}_{t,p})$ is approximately linear (i.e. $f(\mathbf{x}_{t,p}) \approx \mathbf{x}_{t,p}^\top \boldsymbol{\theta}^\star$ for some unobserved $\boldsymbol{\theta}^\star$). The goal of the master algorithm/processors is to utilize its access

to the sequence of rewards $r_{t,p}$ and joint control over the sequence of action selections $\mathbf{x}_{t,p}$ (which depends on the past sequence to events) to minimize the **parallel regret**:

$$\mathcal{R}(T,P) = \sum_{p=1}^{P}\left(\sum_{t=1}^{T} f(\mathbf{x}_{t,p}^{\star}) - f(\mathbf{x}_{t,p})\right),$$

where $\mathbf{x}_{t,p}^{\star} = \arg\max_{\mathbf{x}\in\mathcal{X}_{t,p}} f(\mathbf{x})$. We also introduce the notion of the **best regret** across processors:

$$\mathcal{R}_{*}(T) = \min_{p\in[P]}\left(\sum_{t=1}^{T} f(\mathbf{x}_{t,p}^{\star}) - f(\mathbf{x}_{t,p})\right).$$

which captures the performance of the best processor in hindsight. We note the following relationship which follows immediately by definition of the aforementioned regrets:

Remark 1. The **parallel regret** and **best regret** satisfy,

$$\mathcal{R}_{*}(T) \leq \frac{\mathcal{R}(T,P)}{P}.$$

In the special case that there is a fixed context for all time across all processors (i.e. $\mathcal{X}_{t,p} = \mathcal{X}$), as is the case for design problems, the **simple regret** is useful,

$$\mathcal{R}_{s}(T,P) = f(\mathbf{x}^{\star}) - f(\mathbf{x}_{T+1,1}).$$

The **simple regret** captures the suboptimality of a choice $\mathbf{x}_{T+1}$, given by a next-step policy $\pi(\cdot)$ at the $T+1$st time against the single best choice $\mathbf{x}_{\star} \in \mathcal{X}$.

Remark 2. There exists a randomized next-step policy $\pi(\cdot)$ (depending on the sequence of $\mathbf{x}_{t,p}$) at the $T+1$st timestep such that the simple regret satisfies

$$\mathbb{E}_{\pi}[\mathcal{R}_{s}(T,P)] \leq \mathbb{E}\left[\frac{\mathcal{R}(T,P)}{TP}\right]$$

when there is a fixed global context $\mathcal{X}$ for all $t \in [T]$, $p \in [P]$[1].

For our analysis, we make the following standard assumptions on the bandit instance in (4.1),

Assumption 1 (Subgaussian Noise)**.** *The noise variables $\xi_{t,p}$ are R-subgaussian for all $t \in [T]$ and $p \in [P]$. That is, for every λ*

$$\mathbb{E}[e^{\lambda\xi_{t,p}}] \leq e^{R^2\lambda^2/2}.$$

[1]This follows from a similar reduction from sequential regret to simple regret applicable to multi-armed bandits in [57, Proposition 33.2].

Assumption 2 (Bounded Covariates)**.** *For all contexts $\mathcal{X}_{t,p}$, $t \in [T]$, $p \in [P]$, the actions are norm-bounded by a known upper bound:*

$$\|\mathbf{x}\| \leq L, \quad \forall \mathbf{x} \in \mathcal{X}_{t,p}.$$

Assumption 3 (Almost-Linear Rewards)**.** *The function $f(\cdot)$ is ϵ-close to linear in that for all contexts $\mathcal{X}_{t,p}$ and $\forall \mathbf{x} \in \mathcal{X}_{t,p}$, there exists a parameter $\boldsymbol{\theta}^\star$ such that*

$$|f(\mathbf{x}) - \mathbf{x}^\top \boldsymbol{\theta}^\star| \leq \epsilon.$$

Further, this underlying parameter $\boldsymbol{\theta}^\star$ satisfies the norm bound:

$$\|\boldsymbol{\theta}^\star\| \leq S.$$

In the context of the above conditions we define the signal-to-noise ratio as:

$$\text{SNR} = \left(\frac{LS}{R}\right)^2 \tag{4.2}$$

in analogy with the classical setting of offline linear regression. Note that while we allow ϵ to be arbitrary, our guarantees are only non-vacuous when ϵ is suitably small. Thus $f(\mathbf{x})$ should be thought of a function that is nearly linear. Although we only consider linear algorithms in this paper, we note that our methods can easily be generalized using finite-dimensional feature expansions (i.e. the kernel trick) and random feature approximations [68] to increase the flexibility of our model class.

4.2 Parallelizing Linear Bandits

We consider several algorithms for linear bandits which provably achieve a nearly perfect parallel speed-up. Most of these algorithms are inspired by the *optimism* principle. Such algorithms maintain a running estimate of $\hat{\boldsymbol{\theta}}_t \approx \boldsymbol{\theta}^\star$ and the empirical covariance matrix of queried observations:

$$\mathbf{V}_{t,p} = \lambda \mathbf{I}_d + \sum_{a=1}^{t-1} \sum_{b=1}^{P} \mathbf{x}_{a,b} \mathbf{x}_{a,b}^\top + \sum_{k=1}^{p-1} \mathbf{x}_{t,k} \mathbf{x}_{t,k}^\top$$

which are used to construct a confidence ellipsoid in parameter space where λ is a regularization parameter. The best parameter in this set of plausible parameters–which allows the estimation of a maximum hypothetical reward–is used to optimistically guide exploration in the feature space. We use the notation $\mathcal{F}_{t,p-1}$ to define the filtration of all events that have occurred up until and including the revelation of context $\mathcal{X}_{t,p}$. Despite the topical differences in these algorithms, there is a common thread which ties together their analyses in the parallel setting in our framework[2]:

[2]The choice of 2 in the upper bound here is arbitrary and can be replaced with any universal constant $c > 1$ without changing our results up to constants.

Algorithm 1 Parallel LinUCB

Input: $P, T, R, S, L, \lambda, \epsilon$, DR Routine.

for $t = 1 : T$ **do**
  Compute $\hat{\boldsymbol{\theta}}_t$, $\mathbf{V}_{t,1}$, and β_t as in (4.6) and (4.7).
  for $p = 1 : P$ **do**
    Given $\mathcal{X}_{t,p}$, compute $\mathbf{y}_{t,p} \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}_{t,p}} \max_{\boldsymbol{\theta} \in \mathcal{C}_{t,0}} \mathbf{x}^\top \boldsymbol{\theta}$ for $\mathcal{C}_{t,0}(\hat{\boldsymbol{\theta}}_t, \mathbf{V}_{t,1}, \beta_t, \epsilon)$ as in (4.6).
  end for
  Compute $\tilde{\mathbf{V}}_{t+1,1} = \mathbf{V}_{t,1} + \sum_{p=1}^{P} \mathbf{y}_{t,p}\mathbf{y}_{t,p}^\top$.
  if $\tilde{\mathbf{V}}_{t+1,1} \preceq 2\mathbf{V}_{t,1}$ **then**
    for $p = 1 : P$ **do**
      Set $\mathbf{x}_{t,p} \leftarrow \mathbf{y}_{t,p}$ and query $\mathbf{x}_{t,p}$ to receive reward $r_{t,p}$ } Executed In Parallel
    end for
  else
    Set $\{\mathbf{x}_{t,p}\}_{p=1}^{P} \leftarrow \mathrm{DR}(\mathcal{F}_{t,P})$
    for $p = 1 : P$ **do**
      Query $\mathbf{x}_{t,p}$ to receive reward $r_{t,p}$ } Executed In Parallel
    end for
  end if
end for

Condition 1. [Critical Covariance Inequality] An (estimated) covariance matrix $\mathbf{V}_{t,p} \in \mathcal{F}_{t,p-1}$ is said to satisfy the *critical covariance inequality* at round t for processor p if

$$\boxed{\mathbf{V}_{t,1} \preceq \mathbf{V}_{t,p} \preceq 2\mathbf{V}_{t,1}.} \tag{4.3}$$

We refer to any round t for which the aforementioned inequality does not hold for any $p \in \{2, \ldots, P\}$ as a *doubling round*.

For the standard purely sequential setting of linear bandits (i.e. $P = 1$), in each interaction round we select $\mathbf{x}_t$, gain the information r_t, update our model, and iterate. However, in parallel setting we must jointly select $\mathbf{x}_{t,p}$ for all $p \in [P]$ *before* seeing any additional rewards in round t. Condition 1 ensures that up a factor of 2, there is no direction along which our estimate of the covariance changes too rapidly within a single round. If we imagine unrolling the parallel dimension p sequentially across time (i.e. consider the lexigraphic ordering of pairs (t, p)), Condition 1 ensures the covariance estimate is quasi-static intra-round. The significance of this simple condition, is that once we receive reward $r_{t,p}$ for $p \in [P]$ at the end of the round t, it is nearly as if we received the reward $r_{t,p}$ immediately after selecting $\mathbf{x}_{t,p}$. We later provide more intuition as how this factors into our analysis and what algorithms satisfy this property.

Finally, in the event a particular round t is a doubling round, we allow our algorithms to call a *doubling round routine*, $\{\mathbf{x}_{t,p}\}_{p=1}^{P} \leftarrow \mathrm{DR}(\mathcal{F}_{t,P})$, where as before $\mathcal{F}_{t,P}$ is the σ-

Algorithm 2 Parallel Lazy LinUCB

Input: $P, T, R, S, L, \lambda, \epsilon$, DR Routine.

for $t = 1 : T$ **do**
  Compute $\hat{\boldsymbol{\theta}}_t$, and β_t as in (4.6) and (4.7).
  for $p = 1 : P$ **do**
    Compute $\tilde{\mathbf{V}}_{t,p} = \mathbf{V}_{t,1} + \sum_{k=1}^{p-1} \mathbf{y}_{t,k}\mathbf{y}_{t,k}^\top$.
    Given $\mathcal{X}_{t,p}$, compute $\mathbf{y}_{t,p} \leftarrow \arg\max_{\mathbf{x}\in\mathcal{X}_{t,p}} \max_{\boldsymbol{\theta}\in\mathcal{C}_{t,p}} \mathbf{x}_t^\top \boldsymbol{\theta}$ for $\mathcal{C}_{t,p}(\hat{\boldsymbol{\theta}}_t, \tilde{\mathbf{V}}_{t,p}, \beta_t, \epsilon)$ as in (4.6).
  end for
  Compute $\tilde{\mathbf{V}}_{t+1,1} = \mathbf{V}_{t,1} + \sum_{p=1}^{P} \mathbf{y}_{t,p}\mathbf{y}_{t,p}^\top$.
  if $\tilde{\mathbf{V}}_{t+1,1} \preceq 2\mathbf{V}_{t,1}$ **then**
    for $p = 1 : P$ **do**
      Set $\mathbf{x}_{t,p} \leftarrow \mathbf{y}_{t,p}$ and query $\mathbf{x}_{t,p}$ to receive reward $r_{t,p}$ } Executed In Parallel
    end for
  else
    Set $\{\mathbf{x}_{t,p}\}_{p=1}^P \leftarrow \mathrm{DR}(\mathcal{F}_{t,P})$
    for $p = 1 : P$ **do**
      Query $\mathbf{x}_{t,p}$ to receive reward $r_{t,p}$ } Executed In Parallel
    end for
  end if
end for

algebra containing all information regarding past contexts, rewards, and selected actions. The doubling round routine allows our algorithm to make a different choice of actions instead of the actions $\{\mathbf{y}_{t,p}\}_{p=1}^P$ suggested by the optimistic algorithm if round t is a doubling round. In many cases, this routine can simply be taken to be the *identity* map and non-trivial parallelism gains are still obtained. However, in Section 4.3 we provide an example of a nontrivial choice of doubling round routine that can exploit the geometry of the context sets for improved performance.

Throughout the following sections our regret upper bounds are stated with high-probability. That is, we claim $\mathcal{R}(T, P) \leq \textsc{rate}$ with probability at least $1 - O(\delta)$ where RATE has at most $O(\log(\frac{1}{\delta}))$ dependence on δ. However, under Assumptions 2 and 3, the total regret can always be trivially bounded as $O(LSTP)$. Thus, our high probability regret bounds can be easily converted to upper bounds in expectation at the cost of only logarithmic factors by setting $\delta \propto (\frac{1}{TP})^2$, for example.

Linear Upper Confidence Bound (UCB) Algorithms

We first show how two natural algorithms, which are parallelized variants of the classic LinUCB algorithm of [1], obtain the optimal sequential regret, up to a burn-in term, when

they satisfy Condition 1. In the following we use

$$\hat{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \sum_{a=1}^{t-1} \sum_{b=1}^{P} (r_{a,b} - \mathbf{x}_{a,b}^\top \boldsymbol{\theta})^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \tag{4.4}$$

to refer to the least-squares estimator using data until round t, and

$$\mathcal{C}_{t,p}(\hat{\boldsymbol{\theta}}_t, \mathbf{V}_{t,p}, \beta_t, \epsilon) = \{\boldsymbol{\theta} : \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_t\|_{\mathbf{V}_{t,p}} \leq \sqrt{\beta_t(\delta)} + \sqrt{(t-1)P}\epsilon\} \tag{4.5}$$

$$\text{for } \mathbf{V}_{t,p} = \lambda \mathbf{I}_d + \sum_{a=1}^{t-1} \sum_{b=1}^{P} \mathbf{x}_{a,b} \mathbf{x}_{a,b}^\top + \sum_{k=1}^{p-1} \mathbf{x}_{t,k} \mathbf{x}_{t,k}^\top \qquad \text{where} \tag{4.6}$$

$$\sqrt{\beta_t(\delta)} = R\sqrt{\log\left(\frac{\det(\mathbf{V}_{t,0})}{\lambda^d \delta^2}\right)} + \sqrt{\lambda} S \leq R\sqrt{d \log\left(\frac{1 + tPL^2/\lambda}{\delta}\right)} + \sqrt{\lambda} S \tag{4.7}$$

to refer to a confidence ellipsoid which uses $\hat{\boldsymbol{\theta}}_t$ as its center, but allows the matrix $\mathbf{V}_{t,p}$ (which includes intra-round updates) to modulate the exploration directions. As we will argue, these confidence ellipsoids satisfy the "optimism" property in that they contain the unobserved $\boldsymbol{\theta}^\star$ with high probability. We note that the confidence ellipsoids also include an addition term $\propto \epsilon$ to accommodate the nonlinearity of the objective.

Algorithm 1 exploits parallelism in a simple fashion. It finds the best optimistic upper bound on the reward in round t for each context and allocates each of its P parallel resources to querying those arms. Although this seems redundant (if for example all the context sets are equal), this strategy can still provide benefit because when Algorithm 1 queries a common arm $\mathbf{x}$, P times, the effective variance in the noise of the observed reward is reduced by a factor of $1/P$. As Theorem 2 shows, even this simple parallelism-enabled noise reduction strategy can provide significant benefit.

Algorithm 2 naturally encourages diversity in its parallel exploration strategy within a given round. Within a round t, Algorithm 2 queries new actions $\mathbf{x}_{t,p}$ in the standard LinUCB fashion via an optimistic approach with an important caveat: while the covariance matrix is sequentially updated using the queried actions, a "stale" mean estimate (with data from the first $t-1$ round) is used in the construction of its corresponding confidence ellipsoid since the rewards $r_{t,p}$ are not available intra-round. This update strategy exploits a key property of the linear regression estimator used to construct the confidence ellipsoid. The covariance matrix used to modulate exploration across directions does *not* depend on the rewards $r_{t,p}$ (although the mean estimate does)[3]. One pitfall of such an approach is that shrinking the predicted variance in the absence of corresponding observations can lead to an "overconfident" algorithm which may incorrectly exclude the true parameter $\boldsymbol{\theta}^\star$ from its confidence set. We compensate this aggressive exploration strategy by also using a lazy threshold width $\sqrt{2}\beta_t(\delta)$ which is inflated by a small multiplicative factor to allay this effect.

Our main result follows which bounds the regret of both Algorithm 1 and Algorithm 2.

[3]This is closely related to the fact that the conditional covariance of jointly Gaussian random variable only depends on the covariance of the original matrix.

Theorem 2. *Let Assumptions 1, 2 and 3 hold and $\mathcal{D}_t$ denote the event that round t is a doubling round (see Condition 1). Then, for any choice of doubling-round routine* DR*, the regret of both Algorithms 1 and 2 satisfy,*

$$\mathcal{R}(T,P) \leq O\left(LSP \cdot \sum_{t=1}^{T} \mathbb{1}[\mathcal{D}_t]\right) + \tilde{O}\left(\sqrt{dTP}\max(R\sqrt{d} + \sqrt{\lambda}S + \sqrt{TP}\epsilon, LS)\right) \tag{4.8}$$

with probability at least $1-\delta$.

We now make several comments to interpret the result.

- The second term in Theorem 2 represents the near-optimal (up to log factors) regret a single processor could achieve in a total of TP rounds interacting with a bandit instance in a purely sequential fashion[4]. The first term in Theorem 2 represents the price of parallelization. Theorem 2 hints at the prospect of obtaining a near-optimal worst-case regret (as $T \to \infty$) if the estimate of the covariance can be stabilized intra-round (so the algorithms do not suffer too many doubling-rounds). Perhaps surprisingly we show how the Condition 1 can be enforced such that the first term in Theorem 2 is *independent* of T and d in the sequel. This can even be done in several cases when the choice of the doubling-round routine DR is taken to be the identity map.

- Theorem 2 also explicitly represents the scales of noise, covariates and parameters, and model misspecification (i.e. the values in Assumptions 1, 2 and 3) instead of enforcing the standard normalizations these quantities are $\Theta(1)$ as is standard in the bandits literature (see [57]). Explicitly representing these quantities allows a fine-grained understanding of the interplay between parallelism and quantities such as SNR, which may vary from application to application.

- The cost of misspecification in the regret is high. An ϵ-level of misspecification[5] contributes a linearly-scaling regret of $\epsilon\sqrt{d}TP$ in both T and P. Balancing the trade-off between the variance $\tilde{O}(Rd\sqrt{TP})$ and misspecification bias $\tilde{O}(\epsilon\sqrt{d}TP)$ is more nuanced in the setting of sequential learning compared to that of i.i.d. supervised learning. In particular Theorem 2 suggests that especially at long timescales and large levels of parallelism, the errors compounded by using an inflexible feature set can easily overwhelm the reduced exploration needed when regressing in a low-dimensional space. If large values of P are desired using a flexible feature expansions (with a higher effective dimension) may be desirable.

Section 4.3 provides several sufficient conditions under which the covariance stability can be naturally satisfied. Further, Section 4.4 provides instances showing in the regime of sufficiently large TP the regret bounds in Theorem 2 are unimprovable.

[4]The standard choice of regularizer in the sequential setting is taken as $\lambda = L^2$.

[5]Note this is only non-vacuous when $\epsilon \ll LS/\sqrt{d}$ since the regret can always be trivially bounded by $O(LSTP)$ under our assumptions.

Linear Thompson Sampling (TS) Algorithms

Following Section 4.2, we show two parallel variants of linear Thompson sampling (TS) [5, 4, 2] can obtain optimal sequential regret, up to a burn in-term, when they satisfy Condition 1. Despite being among the oldest bandit methods, and having worse worst-case theoretical guarantees compared to their deterministic counterparts, Thompson sampling methods often achieve excellent performance in practice [88, 73] .

We use the same notation as in Section 4.2 and refer to the least-squares estimator using data until round t as $\hat{\boldsymbol{\theta}}_t$, and defined as in Equation 4.4, to describe our parallel TS variants. Algorithm 3 is the corresponding Thompson sampling version of Algorithm 1. During each round $p \in [P]$, Algorithm 3 samples P independent candidate parameters $\{\tilde{\boldsymbol{\theta}}_{t,p}\}_{p=1}^{P}$ to induce exploration over the parameter set. Solving the optimization problems $\operatorname{argmax}_{\mathbf{x} \in \mathcal{X}_{t,p}} \mathbf{x}^\top \tilde{\boldsymbol{\theta}}_{t,p}$ over the possibly processor dependent contexts $\mathcal{X}_{t,p}$, induces p distinct arm choices for the different processors $\{\mathbf{x}_{t,p}\}_{p=1}^{P}$. While Algorithm 3 does not update the covariance of the sampling distribution while producing each of the P candidate models $\{\tilde{\boldsymbol{\theta}}_{t,p}\}_{p=1}^{P}$, Algorithm 4 mirrors Algorithm 2. Algorithm 4 proceeds by sampling the model parameters sequentially across the different processors, but updates the sampling covariance matrix in between each sampling step intra-round.

Our main result is the following bound for Algorithms 3 and 4:

Theorem 3. *Let Assumptions 1, 2 and 3 hold and $\mathcal{D}_t$ denote the event that round t is a doubling round (see Condition 1). Then, for any choice of doubling-round routine* DR*, the regret of both Algorithms 3 and 4 satisfy,*

$$\mathcal{R}(T, P) \leq O\left(LSP \cdot \sum_{t=1}^{T} \mathbb{1}[\mathcal{D}_t]\right) + \tilde{O}\left(d\sqrt{TP\left(1 + \frac{L^2}{\lambda}\right)}\left(R\sqrt{d} + S\sqrt{\lambda} + \sqrt{TP}\epsilon\right)\right) \quad (4.9)$$

with probability at least $1 - 3\delta$, whenever $\delta \leq \frac{1}{6}$.

We now make several comments on the result,

- Theorem 3 has a similar flavor to Theorem 2–allowing for near perfect parallelization (as $T \to \infty$) relative to the the sequential Thompson sampling algorithm when the first term is independent of T[6]. As before, Condition 1 can be enforced such that the first term in *independent* of T.
- Note that even in the sequential setting, the regret of linear Thompson sampling suffers an extra multiplicative $\sqrt{d}$ factor relative to LinUCB [2], . The parallel variants of Thompson inherit this $\sqrt{d}$ factor as Theorem 3 shows. Despite this extra dimension-dependent factor (which is needed to maintain the optimism property when using the noisily sampled candidate models for exploration) the performance of the Thompson sampling algorithm is often excellent in practice [73] which motivates its study.

[6]The standard bound for TS in the sequential setting can be obtained by setting $\lambda = L^2$.

Algorithm 3 Parallel LinTS

Input: $P, T, R, S, L, \lambda, \epsilon$, DR Routine.

for $t = 1 : T$ **do**
  Compute $\hat{\boldsymbol{\theta}}_t$, $\mathbf{V}_{t,1}$, and β_t as in (4.4), (4.6) and (4.7).
  for $p = 1 : P$ **do**
    Sample $\boldsymbol{\eta}_{t,p} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d)$.
    Compute parameter $\tilde{\boldsymbol{\theta}}_{t,p} = \hat{\boldsymbol{\theta}}_t + \left(\sqrt{\beta_t} + \sqrt{(t-1)P\epsilon}\right) \mathbf{V}_{t,1}^{-1/2} \boldsymbol{\eta}_{t,p}$.
    Given $\mathcal{X}_{t,p}$, compute $\mathbf{y}_{t,p} \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}_{t,p}} \mathbf{x}^\top \tilde{\boldsymbol{\theta}}_{t,p}$.
  end for
  Compute $\tilde{\mathbf{V}}_{t+1,1} = \mathbf{V}_{t,1} + \sum_{p=1}^{P} \mathbf{y}_{t,p} \mathbf{y}_{t,p}^\top$.
  if $\tilde{\mathbf{V}}_{t+1,1} \preceq 2\mathbf{V}_{t,1}$ **then**
    for $p = 1 : P$ **do**
      Set $\mathbf{x}_{t,p} \leftarrow \mathbf{y}_{t,p}$ and query $\mathbf{x}_{t,p}$ to receive reward $r_{t,p}$ } Executed In Parallel
    end for
  else
    Set $\{\mathbf{x}_{t,p}\}_{p=1}^{P} \leftarrow \mathrm{DR}(\mathcal{F}_{t,P})$
    for $p = 1 : P$ **do**
      Query $\mathbf{x}_{t,p}$ to receive reward $r_{t,p}$ } Executed In Parallel
    end for
  end if
end for

4.3 Stable Covariances

We demonstrate how Condition 1 can be naturally satisfied in a variety of settings. The bounds presented here have a common structure which takes the form:

$$\mathcal{R}(T, P) \leq \tilde{O}\left(\underbrace{R\sqrt{\mathrm{SNR}} \cdot P \cdot \kappa}_{\text{burn-in}} + \mathcal{R}(TP, 1)\right) \tag{4.10}$$

where $\mathcal{R}(TP, 1)$ captures the regret of that learning algorithm operating in purely sequential fashion and κ is a geometry-dependent constant. The price of parallelism is factored into the burn-in term. Although this term is subleading as $T \to \infty$, for many applications of interest (such as protein engineering) it may be the case that $P \sim T$ or $P \gg T$. Thus understanding the value of κ, as a function of the context set geometry, is a question of interest.

Arbitrary Contexts

Our first result shows in the general setting of linear contextual bandits, a uniform bound holds on the number of doubling rounds for any sequence of actions selected.

Algorithm 4 Parallel Lazy LinTS

Input: $P, T, R, S, L, \lambda, \epsilon$, DR Routine.

for $t = 1 : T$ **do**
  Compute $\hat{\boldsymbol{\theta}}_t$, and β_t as in (4.6) and (4.7).
  for $p = 1 : P$ **do**
    Compute $\tilde{\mathbf{V}}_{t,p} = \mathbf{V}_{t,1} + \sum_{k=1}^{p-1} \mathbf{y}_{t,k}\mathbf{y}_{t,k}^\top$.
    Sample $\boldsymbol{\eta}_{t,p} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d)$.
    Compute parameter $\tilde{\boldsymbol{\theta}}_{t,p} = \hat{\boldsymbol{\theta}}_t + \left(\sqrt{2\beta_t} + \sqrt{2(t-1)P\epsilon}\right) \mathbf{V}_{t,p}^{-1/2} \boldsymbol{\eta}_{t,p}$.
    Given $\mathcal{X}_{t,p}$, compute $\mathbf{y}_{t,p} \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}_{t,p}} \mathbf{x}^\top \tilde{\boldsymbol{\theta}}_{t,p}$.
  end for
  Compute $\tilde{\mathbf{V}}_{t+1,1} = \mathbf{V}_{t,1} + \sum_{p=1}^{P} \mathbf{y}_{t,p}\mathbf{y}_{t,p}^\top$.
  if $\tilde{\mathbf{V}}_{t+1,1} \preceq 2\mathbf{V}_{t,1}$ **then**
    for $p = 1 : P$ **do**
      Set $\mathbf{x}_{t,p} \leftarrow \mathbf{y}_{t,p}$ and query $\mathbf{x}_{t,p}$ to receive reward $r_{t,p}$ } Executed In Parallel
    end for
  else
    Set $\{\mathbf{x}_{t,p}\}_{p=1}^{P} \leftarrow \mathrm{DR}(\mathcal{F}_{t,P})$
    for $p = 1 : P$ **do**
      Query $\mathbf{x}_{t,p}$ to receive reward $r_{t,p}$ } Executed In Parallel
    end for
  end if
end for

Lemma 1. *Let $\{\mathcal{X}_{t,p}\}_{t=1,p=1}^{T,P}$ be an arbitrary sequence of contexts. If Assumption 2 holds and the covariance is estimated as in (4.7), then almost surely over any sequence $\mathbf{x}_{t,p}$ of selected covariates, the number of total doubling rounds is bounded by at most $\left\lceil \frac{d}{\log(2)} \log\left(1 + \frac{TPL^2}{d\lambda}\right) \right\rceil$.*

Proof. First, note if round t is a doubling round, then there must exist some $\mathbf{v} \in \mathbb{S}^d$ such that $\mathbf{v}^\top \mathbf{V}_{t,P+1}\mathbf{v} > 2\mathbf{v}^\top \mathbf{V}_{t,1}\mathbf{v}$. An application of Lemma 12 shows this implies $\det(\mathbf{V}_{t,P+1}) > 2\det(\mathbf{V}_{t,1})$. So if k doubling rounds elapse by the end of time T it must be the case that $\det(\mathbf{V}_{T,P+1}) > 2^k \det(\mathbf{V}_{1,0}) \implies \log\left(\frac{\det(\mathbf{V}_{T,P+1})}{\det(\mathbf{V}_{1,0})}\right) > k\log(2)$. However by Lemma 11, for any sequence of selected covariates satisfying Assumption 2, we have that $\log\left(\frac{\det(\mathbf{V}_{T,P+1})}{\det \mathbf{V}_{1,0}}\right) \leq d\log\left(1 + \frac{TPL^2}{d\lambda}\right)$. So it follows $k < \left\lceil \frac{d}{\log(2)} \log\left(1 + \frac{TPL^2}{d\lambda}\right) \right\rceil$. □

Lemma 1 ensures in broad generality, the number of doubling rounds bounded by $\tilde{O}(d)$. Instantiating our previous results then gives,

Corollary 2. *In the setting of Theorem 2, choosing $\lambda = L^2$ and taking the doubling-routine* DR *as the identity map, the regret of both Algorithms 1 and 2 satisfy,*

$$\mathcal{R}(T,P) \leq \tilde{O}\left(R \cdot \left(P\sqrt{\text{SNR}} \cdot d + \sqrt{dTP}(\sqrt{d} + \sqrt{\text{SNR}} + \frac{\epsilon}{R}\sqrt{TP})\right)\right)$$

with probability at least $1-\delta$. In the setting of Theorem 3 with the choice $\lambda = L^2$ and also taking the doubling-routine DR *as the identity map, the regret of Algorithms 3 and 4 satisfy,*

$$\mathcal{R}(T,P) \leq \tilde{O}\left(R\left(P\sqrt{\text{SNR}} \cdot d + d\sqrt{TP}(\sqrt{d} + \sqrt{\text{SNR}} + \frac{\epsilon}{R}\sqrt{TP}\right)\right) \tag{4.11}$$

with probability at least $1-3\delta$, whenever $\delta \leq \frac{1}{6}$.

Proof of Corollary 2. Note by Lemma 1, we must have that $\sum_{t=1}^{T} \mathbb{1}[\mathcal{D}_t] \leq \lceil \frac{d}{\log(2)} \log\left(1 + \frac{TPL^2}{d\lambda}\right) \rceil$. An application of Theorems 2 and 3 gives the result after choosing $\lambda = L^2$. □

We can interpret the result as follows.

- The baseline regret of Linear UCB and Lazy Linear UCB interacting in a purely sequential fashion (with the standard choice of regularizer $\lambda = L^2$) for TP rounds scales as,

  $$\mathcal{R}(TP,1) \leq \tilde{O}(R\sqrt{dTP} \cdot (\sqrt{d} + \sqrt{\text{SNR}} + \frac{\epsilon}{R}\sqrt{TP})),$$

  with analogous expression inflated by an extra $\sqrt{d}$ holding for Thompson sampling and Lazy Thompson sampling. The canonical normalizations for the noise, parameter, and covariates in the literature assume $\text{SNR} = R = \Theta(1)$ as well as $\epsilon = 0$, which simplifies to the oft-stated (and optimal) regret $\tilde{O}(d\sqrt{TP})$. In this case, if $T \geq \tilde{\Omega}(P)$, the regret of our parallel algorithms nearly matches the optimal worst-case regret of a single sequential agent.

- The result in Corollary 2 suggests when we opt for a large choice of P, parallelism is particularly beneficial in the small SNR regime. The fact that a low fidelity data-generation process benefits parellelism may seem counter-intuitive. This property arises in part because the algorithms we consider are optimistic – so environments with large SNR have large parameter norms necessitating the usage of large confidence sets which induce more regret.

Arbitrary Contexts with a Stable Initializer

Our next result shows in the general setting of contextual bandits, choosing a large regularizer guarantees a well-conditioned initialization for the empirical covariance that persists for all time.

Lemma 2. *Let Assumption 2 hold. Then almost surely over any sequence of selected $\mathbf{x}_{t,p}$, the covariance estimate in (4.7) satisfies Condition 1 for all $t \in [T]$ if $\lambda = PL^2$.*

Proof. $\mathbf{V}_{t,p} \succeq \mathbf{V}_{t,1}$ follows immediately since $\mathbf{V}_{t,p} - \mathbf{V}_{t,1} = \sum_{k=1}^{p-1} \mathbf{x}_{t,k}\mathbf{x}_{t,k}^\top \succeq 0$. To show the second conclusion, consider an arbitrary vector $\mathbf{v} \in \mathbb{S}^d$. Then $\mathbf{v}^\top \mathbf{V}_{t,p}\mathbf{v} = \mathbf{v}^\top \mathbf{V}_{t,1}\mathbf{v} + \mathbf{v}^\top \sum_{k=1}^{p-1} \mathbf{x}_{t,k}\mathbf{x}_{t,k}^\top \mathbf{v} \leq \mathbf{v}^\top \mathbf{V}_{t,0}\mathbf{v} + pL^2 \leq \mathbf{v}^\top \mathbf{V}_{t,1}\mathbf{v} + \mathbf{v}^\top(\lambda \mathbf{I})\mathbf{v} \leq 2\mathbf{v}^\top \mathbf{V}_{t,1}\mathbf{v}$ under the setting of the result, which gives the conclusion. □

Lemma 2 shows with appropriate choice of regularizer, *no round* is ever a doubling round for the standard estimate of the empirical covariance. Combining with our previous guarantees gives,

Corollary 3. *In the setting of Theorem 2, if the regularizer is chosen as $\lambda = PL^2$ and the doubling-routine* DR *taken as the identity map, then the regret of Algorithms 1 and 2 satisfy,*

$$\mathcal{R}(T, P) \leq \tilde{O}\left(R\sqrt{dTP} \cdot \left(\sqrt{d} + \sqrt{\text{SNR}}\sqrt{P} + \frac{\epsilon}{R}\sqrt{TP}\right)\right)$$

with probability at least $1-\delta$. In the setting of Theorem 3 if $\lambda = PL^2$ and the doubling-routine DR *taken as the identity map, the regret of Algorithms 3 and 4 satisfy,*

$$\mathcal{R}(T, P) \leq \tilde{O}\left(Rd\sqrt{TP}\left(\sqrt{d} + \sqrt{\text{SNR}}\sqrt{P} + \frac{\epsilon}{R}\sqrt{TP}\right)\right)$$

with probability at least $1 - 3\delta$, whenever $\delta \leq \frac{1}{6}$.

Proof of Corollary 3. The result follows by combining Lemma 2 with Theorems 2 and 3 since no round is a doubling round for all $t \in [T]$. □

We now make several comments on the result.

- A large regularizer ensures a stable covariance estimate (and no doubling rounds), but comes at the cost of introducing additional bias (depending on P) into the least-squares estimate. Such bias factors into the second term of the regret since it must be compensated for by using wider confidence sets in the algorithms. Moreover, such a bias results in error that scales multiplicatively in regret as $\sqrt{T}$.
- For $P \cdot \text{SNR} \leq \tilde{O}(d)$, by Corollary 3, the regret of our parallel algorithms nearly matches the optimal worst-case regret of a corresponding single sequential agent.

Finite Context Sets

Next we show how structure in the context set (in this case, finiteness of the action space) can be leveraged to bound the number of doubling rounds without modifying the standard choices of the hyperparameters for the optimistic algorithms considered here.

Lemma 3. *Let $\mathcal{X}_{t,p} \subset \mathcal{X} = \{\mathbf{x}_i\}_{i=1}^m$ for all $t \in [T], p \in [P]$ where $|\mathcal{X}| = m$ is a finite set of vectors. If Assumption 2 holds and the covariance is estimated as in (4.7), then almost surely over any sequence $\mathbf{x}_{t,p}$ of selected covariates, the number of total doubling rounds is bounded by at most $m \cdot \log_2(\lceil P \rceil)$.*

Proof. We first recall from Condition 1 that in a doubling round at time t

$$\mathbf{V}_{t,P+1} \not\preceq 2\mathbf{V}_{t,1}. \tag{4.12}$$

If $\mathcal{X}_{t,p} \subset \mathcal{X} = \{\mathbf{x}_i\}_{i=1}^m$ where $|\mathcal{X}| = m$ is a finite set of vectors, then

$$\mathbf{V}_{t,1} = \lambda \mathbf{I}_d + \sum_{i=1}^{m} w_{t,1}(i)\mathbf{x}_i\mathbf{x}_i^\top$$

Where $w_{t,1}(i)$ corresponds to the number of times action $\mathbf{x}_i$ has been played by all processors up to and including all P actions played at time $t-1$. Whenever t is a doubling round and (4.12) holds, there must exist $i \in [m]$ such that

$$w_{t,P+1}(i) > 2w_{t,1}(i). \tag{4.13}$$

since otherwise for all i it would hold that $w_{t,P+1}(i) \leq 2w_{t,1}(i)$ implying that $\mathbf{V}_{t,P+1} \preceq 2\mathbf{V}_{t,1}$ contradicting (4.12). Observe that each time (4.13) holds, $w_{t,P+1}(i) - w_{t,1}(i) > w_{t,1}(i)$, implying that during round t arm i was pulled more times than the total number of times it has been pulled thus far. Then, for all $i \in [m]$, the difference $w_{t,P+1}(i) - w_{t,1}(i) \leq P$ and therefore for any $i \in [m]$ condition (4.13) cannot hold for more than $\lceil \log_2(P) \rceil$ iterations. Since there are only m underlying vectors in the contexts the result follows. □

This observation implies a bound for our parallel bandit algorithms over finite context sets:

Corollary 4. *In the setting of Theorem 2, assume the context sets $\mathcal{X}_{t,p} \subset \mathcal{X}$ for all $t \in [T], p \in [P]$ where $|\mathcal{X}| = m$ is a finite set of vectors. Then choosing $\lambda = L^2$ and taking the doubling-routine* DR *as the identity map, the regret of both Algorithms 1 and 2 satisfy,*

$$\mathcal{R}(T,P) \leq \tilde{O}\left(R \cdot \left(P\sqrt{\text{SNR}} \cdot m + \sqrt{dTP}(\sqrt{d} + \sqrt{\text{SNR}} + \frac{\epsilon}{R}\sqrt{TP})\right)\right)$$

with probability at least $1-\delta$. In the setting of Theorem 3 with the choice $\lambda = L^2$ and also taking the doubling-routine DR *as the identity map, the regret of Algorithms 3 and 4 satisfy,*

$$\mathcal{R}(T,P) \leq \tilde{O}\left(R\left(P\sqrt{\text{SNR}} \cdot m + d\sqrt{TP}(\sqrt{d} + \sqrt{\text{SNR}} + \frac{\epsilon}{R}\sqrt{TP}\right)\right) \tag{4.14}$$

with probability at least $1-3\delta$, whenever $\delta \leq \frac{1}{6}$.

Proof of Corollary 4. By Lemma 3, we have that $\sum_{t=1}^{T} \mathbb{1}[\mathcal{D}_t] \leq m\lceil \log_2(P) \rceil$. An application of Theorems 2 and 3 gives the result after choosing $\lambda = L^2$. □

We now interpret this result.

- As before, the second terms in Corollary 4 captures the effect of perfect parallel speed-up–it is the regret that a single agent would achieve playing for a total of TP rounds while the first term in Corollary 4 is a burn-in term that bounds the number of rounds which may not be doubling rounds.

- Relative to Corollary 2 the scaling on the burn-in term $\propto P$, contains a factor m instead of d. Thus, if $m \ll d$, Corollary 4 shows how the algorithms considered here can take advantage of additional structure in the context set to mitigate the the cost of parallelism.

- The Proof of Corollary 4 does not exploit any correlation structure in the action set to bound the number of doubling rounds–for example, clusters of arms that are tightly bunched together in the global context space under a suitable notion of distance. Under natural conditions on the action set, sharper instance-dependent bounds may be possible.

Rich Context Sets

Finally we consider a setting where we are presented with a sequence of context sets $\mathcal{X}_{t,p}$ with regularity structure we formally define as follows,

Definition 2. The contexts $\mathcal{X}_{t,p}$ and distributions $\pi_{t,p}(\cdot)$ are a pair of *rich exploration contexts/distributions* if there exists χ^2, $\pi^2_{\max}$ and $\pi^2_{\min}$ such that,

- $\mathbf{x} \sim \pi_{t,p}(\cdot)$ satisfies $\mathbf{x} \in \mathcal{X}_{t,p}$ almost surely for all $t \in [T]$, $p \in [P]$.

- For any sequence $\mathbf{x}_{t,p} \in \mathcal{X}_{t,p}$ selected by the bandit algorithm for $t \in [T]$, $\sum_{p=1}^{P} \mathbf{x}_{t,p}\mathbf{x}_{t,p}^\top \leq P\chi^2\mathbf{I}$.

- Given $\mathbf{x}_{t,p} \sim \pi_{t,p}(\cdot)$, with population mean and covariance defined as, $\mathbb{E}[\mathbf{x}_{t,p}] = \mu_{t,p}$ and $\mathbb{E}[(\mathbf{x}_{t,p} - \mu_{t,p})(\mathbf{x}_{t,p} - \mu_{t,p})^\top] = \mathbf{\Sigma}_{\pi_{t,p}}$, we have that $\pi^2_{\max}\mathbf{I} \succeq \mathbf{\Sigma}_{\pi_{t,p}} \succeq \pi^2_{\min}\mathbf{I}$ for all $t \in [T], p \in [P]$.

Intuitively, Definition 2 guarantees the sequence of presented contexts are (1) sufficiently similar since there is a common p.s.d. ordering for the covariances across all contexts and (2) sufficiently cover all directions in $\mathbb{R}^d$ when the parameters χ^2, $\pi^2_{\max}$ and $\pi^2_{\min}$ are of the same order. If these conditions are satisfied then exploration distributions exist which can uniformly explore all directions of the underlying context sets well. We now provide a simple example of a set of stochastically generated contexts which obey Definition 2. For each processor, let there be a single context set randomly generated as $\mathcal{X}_p = \{\mathbf{x} : \mathbf{x} \sim \mathcal{D}(\cdot)\}_{i=1}^m$ where $\mathcal{D}(\cdot)$ is a $O(1)$-subgaussian and $O(1)$-bounded distribution in $\mathbb{R}^d$ such that $\mathcal{X}_{t,p} = \mathcal{X}_p$ for all $t \in [T]$, $p \in [P]$. Now define $\pi_{t,p}(\cdot)$ as the uniform distribution over all the vectors in a given context $\mathcal{X}_p$ at time t. Then using a simple matrix concentration argument, we

Algorithm 5 Random Exploration Subroutine

Input: $\pi_{t,p}(\cdot)$.

1: **for** $p = 1 : P$ **do**
2: Sample $\mathbf{x}_p \sim \pi_{t,p}(\cdot)$
3: **end for**
4: **return** $\{\mathbf{x}_p\}_{p=1}^P$

can verify there exists $\chi^2 \leq \tilde{O}(1/d)$, $\pi^2_{\max} \leq \tilde{O}(1/d)$ and $\pi^2_{\min} \geq \tilde{\Omega}(1/d)$ in Definition 2 when $P \geq \tilde{\Omega}(d)$ (with probability at least $1-\delta$ over the randomness in $\mathcal{D}(\cdot)$ and $\pi(\cdot)$).

Sampling from a rich exploration policy (when it exists) can serve as effective doubling-round subroutine, since it can help stabilize the covariance intra-round in later rounds. Random exploration helps stabilize the covariance in later rounds as a consequence of concentration: given a set of randomly sampled covariates $\{\mathbf{x}_{i,p}\}_{i=1,p=1}^{N,P}$, we expect $\frac{1}{NP}\sum_{j=1}^{N}\sum_{p=1}^{P}\mathbf{x}_{i,p}\mathbf{x}_{i,p}^\top \approx \mathbf{\Sigma}_\pi$, where $\mathbf{\Sigma}_\pi \approx \mathbf{\Sigma}_{\pi_{t,p}}$ for all $t \in [T], p \in [P]$. So if a significant number of doubling rounds occur (during which Algorithm 5 is used as DR), then later rounds are unlikely to be doubling rounds since the covariance matrix will have a large component proportional $\pi^2_{\min}\mathbf{I} \preceq \mathbf{\Sigma}_\pi$ in its spectrum.

Using this idea we obtain,

Corollary 5. *In the setting of Theorem 2, assume Definition 2 holds for some rich exploration policies, $\{\pi_{t,p}(\cdot)\}_{t=1,p=1}^{T,P}$, and context pairs, $\{\mathcal{X}_{t,p}\}_{t=1,p=1}^{T,P}$. Then choosing $\lambda = L^2$ and taking the doubling-routine* DR *as Algorithm 5 with these $\pi_{t,p}(\cdot)$, the regret of both Algorithms 1 and 2 satisfy,*

$$\mathcal{R}(T,P) \leq \tilde{O}\left(R\cdot\left(\sqrt{\text{SNR}}\cdot\left(\frac{L^2\pi^2_{\max}}{\pi^4_{\min}} + P\frac{\chi^2}{\pi^2_{\min}}\right) + \sqrt{dTP}(\sqrt{d} + \sqrt{\text{SNR}} + \frac{\epsilon}{R}\sqrt{TP})\right)\right)$$

with probability at least $1-2\delta$. In the setting of Theorem 3 with the choice $\lambda = L^2$ and also taking the doubling-routine DR *as Algorithm 5 with this $\pi_{t,p}(\cdot)$, the regret of Algorithms 3 and 4 satisfy,*

$$\mathcal{R}(T,P) \leq \tilde{O}\left(R\left(\sqrt{\text{SNR}}\cdot\left(\frac{L^2\pi^2_{\max}}{\pi^4_{\min}} + P\frac{\chi^2}{\pi^2_{\min}}\right) + d\sqrt{TP}(\sqrt{d} + \sqrt{\text{SNR}} + \frac{\epsilon}{R}\sqrt{TP})\right)\right)$$

with probability at least $1-4\delta$, whenever $\delta \leq \frac{1}{6}$.

We can provide further intepretation of this result as follows.

- The guarantee presented here resembles that of Corollary 4. However, here the coefficient on the burn-in term (which bounds the number of doubling rounds) scales as $\frac{L^2\pi^2_{\max}}{\pi^4_{\min}} + P\frac{\chi^2}{\pi^2_{\min}}$ and is valid even for infinite context sets[7]. The scaling with $\frac{\chi^2}{\pi^2_{\min}}$ is

[7]Recall the coefficient in Corollary 4 scales as m where is a bound on the number of distinct vectors in the presented contexts.

natural. Quantities proportional to $\frac{\chi^2}{\pi^2_{\min}}$ represent the cost of the non-homogeneous geometry of the presented contexts.

- In contrast to Corollaries 2, 3 and 4, Corollary 5 takes advantage of a common geometric structure in the presented contexts $\mathcal{X}_{t,p}$. Corollary 4 is applicable to general context sets (for which we may have $\pi^2_{\min} \approx 0$). However, if $m = \Theta(\exp(d))$ for example, the guarantee degrades badly. Similarly, Corollary 2 has a burn-in term scaled by d. If the context sets are well-conditioned, in the sense that $\frac{\chi^2}{\pi^2_{\min}} \ll d$, then the parallelism cost here is significantly lower then in the aforementioned cases.

- In the example of rich context sets/distributions pair presented in the text, we can have $\frac{L^2\pi^2_{\max}}{\pi^4_{\min}} + P\frac{\chi^2}{\pi^2_{\min}} \leq \tilde{O}(P)$ with high probability when $P \gtrsim \tilde{\Omega}(d)$. As $P \to \infty$, the burn-in term only scales as $\tilde{O}(P)$, which up to logarithmic factors, is free of any explicit dependence on the size of the context sets.

4.4 Parallel Regret Lower Bounds

Lastly, we show that the upper bounds on the parallel regret provided in the previous sections are nearly matched by complementary information-theoretic lower bounds on the parallel regret under natural parameter scalings. Our regret lower bounds seek to capture two specific terms. The first is the regret induced by the learning and optimally playing $\boldsymbol{\theta}^\star$ for a well-specified model and the second is the cost of misspecification. Together this is captured in,

Theorem 4. *For any parallel bandit algorithm, there exists bandit environments satisfying Assumptions 1, 2 and 3,*

1. *such that when $\epsilon = 0$ there is a single global context set (i.e. $\mathcal{X} = \mathcal{X}_{t,p}$ for all $t \in [T], p \in [P]$) for which,*

$$\mathbb{E}[\mathcal{R}(T,P)] \geq \Omega\left(Rd\sqrt{TP}\right) \tag{4.15}$$

   when $T \geq d\max(1, \frac{1}{3\sqrt{2}\sqrt{\text{SNR}}})$,

2. *and a single (finite) global context set with m vectors such that when $d = \lceil 8\log(m)/\epsilon^2 \rceil$, $R = 0$ and $LS \geq 1$,*

$$\mathbb{E}[\mathcal{R}(T,P)] \geq \Omega\left(\epsilon\sqrt{\frac{d-1}{\log(m)}} \cdot \min(TP, m-1)\right). \tag{4.16}$$

The proof of Theorem 4 reposes on three separate parts. The first component is a reduction which argues the minimax regret of a parallel bandit algorithm, presented with contexts fixed across processors, must be at least as much as its sequential counterpart given

access to the same number of total arm queries. The first part of Theorem 4 then follows from a standard construction for lower bounding the sequential regret of a bandit instance when the context set is taken to be a sphere. The second part is a (noiseless) lower bound which uses a probabilistic argument to witness a finite context set, function f, and $\boldsymbol{\theta}^\star$ for which the misspecification level must show up multiplicatively in the regret. We now make several comments to further interpret these results.

- Together the terms (4.15) and (4.16) show the main components of Theorem 2 are unavoidable – in particular the term corresponding to the variance of learning $Rd\sqrt{TP}$ and the term capturing the magnitude of misspecification $\epsilon\sqrt{d}TP$.

- The first term (4.15) in Theorem 4, captures the variance of learning under optimism in the parallel regret. For this portion of the lower bound, the context set is taken to be the sphere which satisfies the conditions of a rich exploration set when the exploration distribution is taken as the uniform distribution over the spherical shell (here $\ell = L$). Hence for $\text{SNR} \lesssim d$ and $\epsilon = 0$, the guarantee from Corollary 5 matches this lower bound up to logarithmic factors for sufficiently large T. Similarly in the regime $P \lesssim d$, the guarantee from Corollary 3 which holds in the absence of any additional structure on the context sets, also matches this lower bound up to logarithmic factors when $\epsilon = 0$.

- To gain further intuition for (4.16) it is helpful to consider the high-dimensional scaling limit where $m = \Theta(d^k)$ for $k \gg 1$ and $TP \ll m$. Then under the conditions of the result, we can see for any ϵ there exists a sufficiently large d so that the parallel regret satisfies $\geq \tilde{\Omega}(\epsilon\sqrt{d}TP)$. Hence (in the realistic regime) where the context set contains large numbers of context vectors and there are not sufficiently many queries to observe all m of them, the $\epsilon\sqrt{d}TP$ in the regret within Theorem 2 is unavoidable up to logarithmic factors.

- Capturing the necessity of the "burn-in" terms, which represent the price of parallelism in our upper bounds, is an interesting but challenging research direction. In particular, because in many applications the information-theoretic limits of learning when $P \sim T$ may be of interest. However, the interplay between the structure of the context set and the burn-in terms in the upper bounds in Corollaries 2, 3, 4 and 5 seems quite nuanced[8]. Any such lower bounds capturing these dependencies will likely need to be constructed on a case-specific basis for different context set geometries as well as be geared towards the small T regime.

[8]Note that an additive dependence of $\Omega(P)$ must at least be necessary, since at time $T = 1$ the algorithm must make P queries to an arbitrary set of contexts in the absence of any information about the underlying bandit environment.

4.5 Experiments

Here we explore the performance of the parallel linear bandit algorithms presented in this paper on several synthetic and real problem instances of increasing complexity. While analysis in the bandits literature is often focused on minimizing regret, in the batch setting best arm identification may be of primary interest for some practical design settings where no cost is incurred for additional arms after the best performing arm within a round. Hence, we explore the performance of our family of algorithms in both parallel regret and best arm identification.

In the synthetic data settings, we investigate both a perfectly linear setting and a misspecified setting generated from the output of a randomly initialized neural network. The real data instances are derived from a material science and biological sequence design applications to provide breadth across a variety of context set geometries and application-specific behaviors. In the real data settings, we consider the performance over parallel variants of all algorithms considered herein against a baseline which is the ϵ-greedy algorithm. Note that this baseline makes no structural assumptions on the conditional model $y_i|x_i$ (and as such is "unbiased") but also is not able to take advantage of the covariates $\mathbf{x}_i$, since it doesn't construct a regression model to guide exploration.

For all experimental setups, we fix the total number of arm queries TP and run with 3 different levels of parallelism (i.e. $P = \{1, 10, 30\}$ for the superconductor setting and $P = \{1, 10, 100\}$ for all others) over 30 separate trials. All algorithms use a doubling round routine which is set to the identity map. As is common, for both Thompson sampling variants, we avoided inflating the confidence set radius by the additional $\sqrt{d}$ factor so it matches the confidence sets of the other algorithms. The misspecification parameter was set to 0 for all experiments. The hyperparameters of each algorithm were tuned via a post-hoc grid search over a logarithmically-spaced grid for the random neural network and real data experiments (see Section 4.7 for details) as in [30].

Synthetic Experiments

We begin by testing the ability of our optimistic algorithms to parallelize on simulated data. We consider a problem in $d = 100$ with a linear reward oracle whose underlying parameter $\boldsymbol{\theta}^\star/\|\boldsymbol{\theta}^\star\|$ for $\boldsymbol{\theta}^\star \sim \mathcal{N}(0, \mathbf{I}_d)$ subject to Gaussian additive noise $\epsilon \sim \mathcal{N}(0, 1)$. We then generate a fixed, global context set $\mathcal{X} = \{\mathbf{x}_i/\|\mathbf{x}_i\|_2\}_{i=1}^m$ for $\mathbf{x}_i \sim \mathcal{N}(0, \mathbf{I}_d)$ with $m = 10^4$ actions. We then set $\mathcal{X}_{t,p} = \mathcal{X}$ for all $t \in [T]$ and $p \in [P]$. The hyperparameters of the algorithms were chosen according to their theoretically-motivated values $\lambda = 1, R = 1, S = 1$ with $\delta = 1/T$. As Fig. 4.1 shows, the parallel versions of each of the algorithms asymptotically achieve a nearly perfect speed-up with respect to parallelism as measured by the regret. As $T \to \infty$ the performance of the different types of base algorithms are comparable.

Next we investigate the parallelism of our methods under changing context sets. We generate a random context set $\mathcal{X}_{t,p} = \{\mathbf{x}_i/\|\mathbf{x}_i\|_2\}_{i=1}^m$ for $\mathbf{x}_i \sim \mathcal{N}(0, \mathbf{I}_d)$ with $m = 10^4$ actions for each timestep-processor pair (t, p) with hyperparameters set as before. Once again we

see in Fig. 4.2 that each algorithm achieves near perfect speed-up as measured by parallel regret and all base algorithms are asymptotically comparable.

Finally, from our theoretical results we recall the importance of the covariances $\mathbf{V}_{t,p}$ remaining quasi-static intra-round. We examine this behavior in the synthetic linear reward setting by determining the minimal doubling round coefficient $\alpha_{t,p}^{\min}$ for each arm query which satisfies $0 \preceq \alpha\mathbf{V}_{t,1} - \mathbf{V}_{t,p}$, we call α the *doubling round coefficient*. That is, for any doubling round coefficient $\alpha \geq \alpha_{t,p}^{\min}$ for all $p \in [P]$, the critical covariance inequality is satisfied and no doubling round is triggered for that round. Note that in our theoretical results, we arbitrarily set $\alpha = 2$ for our analysis. As before, we generate a fixed global context and theoretically-motivated hyperparameter values, but this time with $d = 20$ and $m = 10^3$. We then run the algorithms without calling any doubling round routines and observe how the doubling round coefficient changes through time for $P = 100$. At each (t, p) timestep, we compute $\alpha_{t,p}^{\min}$ and plot it against the number of arm queries as shown in Fig. 4.3. As expected, we qualitatively notice a sawtooth pattern in all algorithms as the minimal doubling round coefficient increases for fixed t as the covariance $\mathbf{V}_{t,p}$ gets updated for each additional processor p which then resets back down at the end of each round. Additionally, all algorithms experience a dropoff in minimal doubling round coefficient with each successive round indicating that the covariance gets more stable as more arms are queried with all algorithms having an essentially flat $\alpha_{t,p}^{\min}$ near 1 as $t \geq 2000$. Finally, we observe that in the fixed context setting where diversity of arms isn't introduced via the context, LinUCB has the highest doubling round coefficients since the algorithm chooses the same arm for all processors leading for significant changes in the shape of the covariance within a round making the algorithm susceptible to overconfidence. Note that the other 3 algorithms have much smaller doubling round coefficients for a much shorter time due to the increased diversity in actions played in the bandit algorithm.

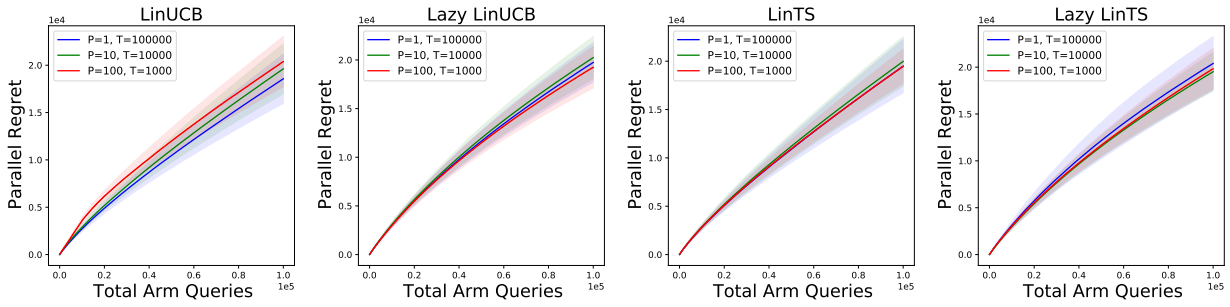

Figure 4.1: Fixed context setting. From left to right: Regret of LinUCB, Lazy LinUCB, LinTS, and Lazy LinTS for varying values of P. The mean regret is plotted across 30 runs with the standard deviation as the shaded region. Here $d = 100$, $m = 10^4$.

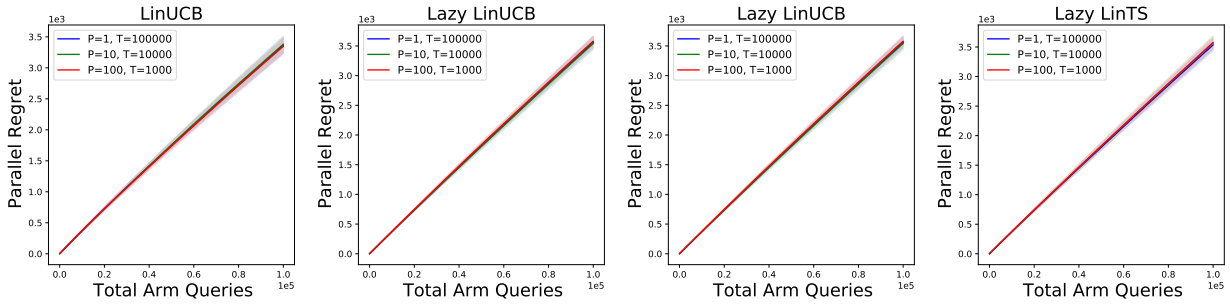

Figure 4.2: Changing context setting. From left to right: Regret of LinUCB, Lazy LinUCB, LinTS, and Lazy LinTS for varying values of P. The mean regret is plotted across 30 runs with the standard deviation as the shaded region. Here $d = 100$, $m = 10^4$.

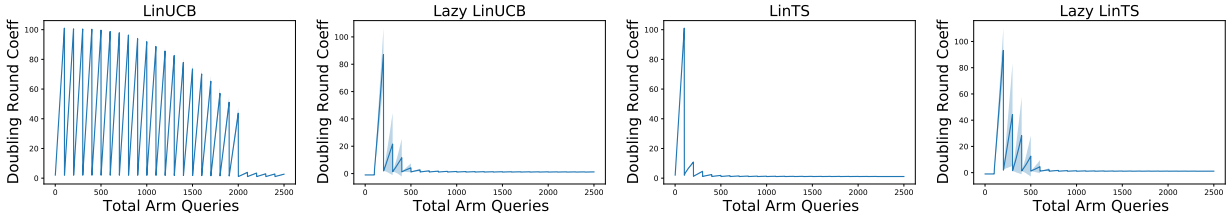

Figure 4.3: Doubling round coefficients. From left to right: doubling round coefficients of LinUCB, Lazy LinUCB, LinTS, and Lazy LinTS. The mean coefficient is plotted across 30 runs with the standard deviation as the shaded region and $d = 20$, $m = 10^3$, and $P = 100$.

Randomly Initialized Neural Network Data

Recent studies ([7], [18]) have modeled fitness landscapes from biological sequence design problems with randomly initialized neural networks as both share many statistical properties in common. Furthermore, randomly initialized neural network exhibit nearly linear properties which are essential to the guarantees for our algorithms as demonstrated in the prequel while deviating from a linear model enough to serve as a good testbed for model misspecification. The "biological sequence" input is modeled as a 14-length binary string $x_i \in \{0,1\}^{14}$ ($m = 16,384$ sequences) to mimic the combinatorial nature of biological sequences. The fitness landscape $f(x_i)$ is modeled by a feedforward neural network with 3 hidden layers (of size 128, 256, and 512 hidden units) where each weight is i.i.d. sampled via Xavier initialization $w \sim \text{Unif}(-\sqrt{6/(h_i + h_{i+1})}, \sqrt{6/(h_i + h_{i+1})})$ where h_i is the number of units in layer i. The output y_i of the randomly initialized feedforward neural network can be thought of as the oracle fitness landscape which we wish to optimize. Note unlike in common use cases for neural networks the initialized weights are never modified. To model experimental noise, we add Gaussian noise to generate the reward $r_i = y_i + \epsilon_i$ where

$\epsilon_i \sim \mathcal{N}(0, 0.5^2)$.

Our family of bandit algorithms were run with both linear features only ($d = 14$) and quadratic features ($d = 210$). In this setting, we first verify that a linear model is appropriate. The best fit linear model for linear features and quadratic features had an R^2 of 0.7 and 0.87, respectively.

As Fig. 4.4 demonstrates the quadratic feature setting, the variance between runs of the parallel regret for the lazy methods tends to be much higher than the non-lazy methods due to the correlation of covariance updates within a batch. LinUCB performs the best upfront in the purely sequential setting due to the high early round cost that Thompson sampling pays upfront while paying a much lower price in regret in successive rounds. In higher parallelism regimes, Thompson sampling performs the best in terms of parallel regret after the first few hundred arm queries. This indicates that Thompson sampling benefits from encouraging diversity.

In comparison as shown in Fig. 4.8, the linear features perform demonstrably worse in terms of parallel regret than the quadratic features in just a few hundred arm queries across all levels of parallelization. Furthermore, the performance of LinUCB in the linear feature setting suffers most significantly relative to the other methods further confirming that diversity is important in settings of model misspecification.

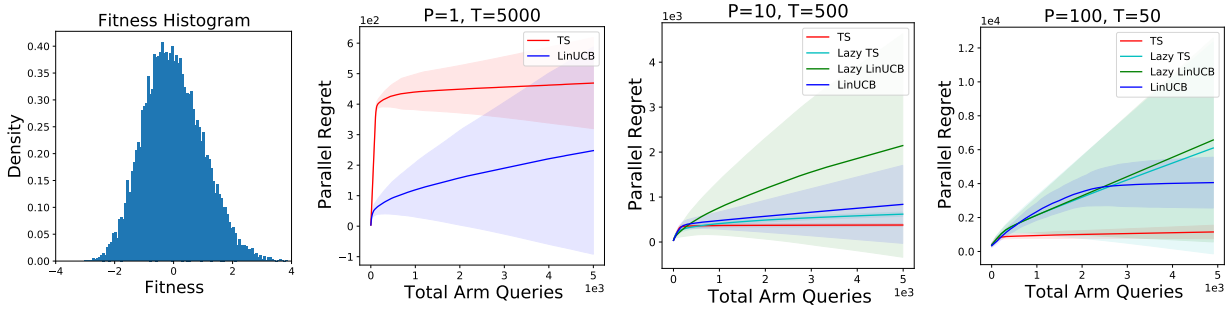

Figure 4.4: Top Left: The histogram of fitness values for the RandomNN dataset. Top right: The parallel regret of the purely sequential setting for 5000 queries with a noise standard deviation of 0.5. Bottom Left: The parallel regret for $P = 10$. Bottom Right: The parallel regret for $P = 100$. The mean regret and standard deviation are plotted as the solid line and shaded region in all plots.

Superconductor Data

To assess the utility of the parallel bandit algorithms in a realistic setting we constructed a semi-synthetic problem using the UCI dataset in [35] consisting of a collection of superconducting materials along with their maximum superconducting temperature. The dataset consists of $m = 21,263$ superconducting materials, each with a $d = 81$-dimensional feature

vector, x_i, containing relevant attributes of the materials chemical constituents and a superconducting critical temperature y_i. We construct a finite-armed bandit oracle over the m arms which returns a reward $r_i = y_i + \epsilon_i$ for $\epsilon_i \sim \mathcal{N}(0, 100^2)$ (since $\max_i y_i = 185.0$). The task in this example is to find the best (highest temperature) superconducting material (or arm as measured by y_i) given access to a total number of arm queries $\ll m$.

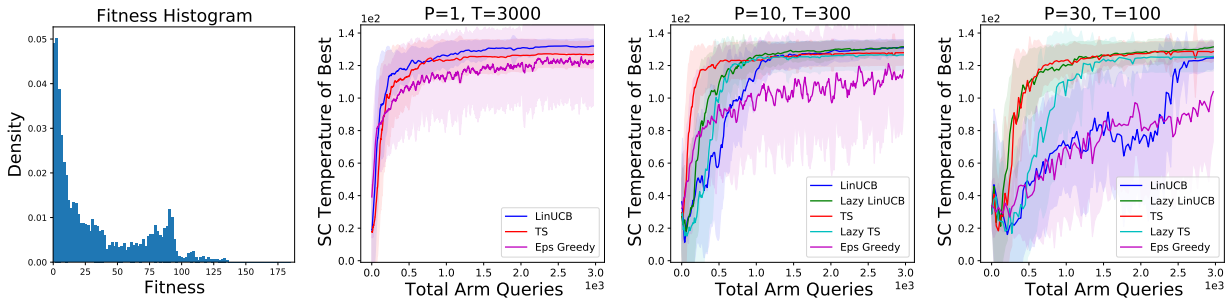

Figure 4.5: Leftmost: Fitness Histogram of Landscape. Left to right: Regret of all algorithms for $P = 1, 10,$ and 30, respectively. Here the best superconducting material (by temperature) as determined by the algorithm at the time is displayed. Curves are also smoothed by a moving-average over a window of size 30 for clarity.

As Fig. 4.5 shows, although ϵ-greedy is a simple algorithm, it can achieve reasonable performance (at the cost of high variance), when $P = 1$. Indeed prior work has shown that other greedy (linear bandit) algorithms are formidable baselines in setting with diverse covariates [12]. However, in our setting, it is still outperformed by all the linear bandit algorithms studied herein. We also see all algorithms quickly saturate to find superconducting materials with temperatures $y_i \approx 120$.

In the cases of $P = 10$, and $P = 30$ we see all the parallel variants of the linear algorithms studied herein achieve non-trivial parallelism gains; that is the number of sequential rounds needed to discover this best material does not scale linearly with P for any of the methods. Remarkably, Thompson sampling suffers almost no loss in performance even when $P = 30$ in this setting with real data where model misspecification is in full force. Thompson sampling outperforms all other algorithms when $P = 10$ and $P = 30$. As our results show, explicitly introducing diversity into the selection of actions provides value in this setting.

Transcription Factor Binding

In order to evaluate the effectiveness of the family of proposed parallelized linear bandit algorithms in a realistic biological sequence design setting, we utilized a fully characterized experimental transcription factor binding affinity dataset from [11] (using the software package in [83]). Changes in transcription factor binding affinity has been shown to have impact on gene regulatory function and subsequently is associated with disease risk. The dataset

experimentally characterizes the binding affinity of all possible length-8 DNA sequence motifs ($m = 4^8 = 65,536$) to a transcription factor DNA binding domain providing a good benchmark for our bandit methods in a real biological application with the combinatorial structure common in biological sequence design. In this setting the number of arms m is $O(\exp(d))$.

Often in biological sequence design problems quadratic features are used to model pairwise interactions (referred to as epistasis in biology). We compared linear features with random ReLU features and a quadratic kernel and found linear features work best in and provide additional insight in Section 4.7.

The fully characterized landscape allows for exact computation of parallel regret and analysis of the impact of realistic forms of model misspecification. Each arm x_i was one-hot encoded. The scaled binding affinity $y_i \in [0, 1]$ measured the binding of the arm x_i for the SIX6 REF R1 transcription factor target. The finite-armed bandit oracle as in the superconductor setting was modeled as $r_i = y_i + \epsilon_i$ where $\epsilon_i \sim \mathcal{N}(0, 0.3^2)$. The task for this application is to find the sequence with the highest transcription factor binding affinity.

In evaluating the best arm reported across varying levels of parallelism, we can see that LinUCB and ϵ-greedy consistently performs the worst with the other 3 algorithms(Lazy LinUCB, Thompson sampling, and Lazy Thompson sampling) perform comparably. This implies that diversity of arms is important particularly when the model is misspecified. Note that the right tail of the fitness histogram is rather heavy for this task such that getting to an arm with fitness above 0.9 is rather simple in the noiseless setting and can be optimized in few arm queries as was shown in [7]. However, since biological experiments often have large experimental error we add noise with standard deviation of 0.3 making the problem significantly harder leading to worse performance of algorithms preventing ϵ-greedy from beating the 0.9 threshold at all. Similarly, we find the same relative performance of methods in terms of parallel regret as shown in Section 4.7.

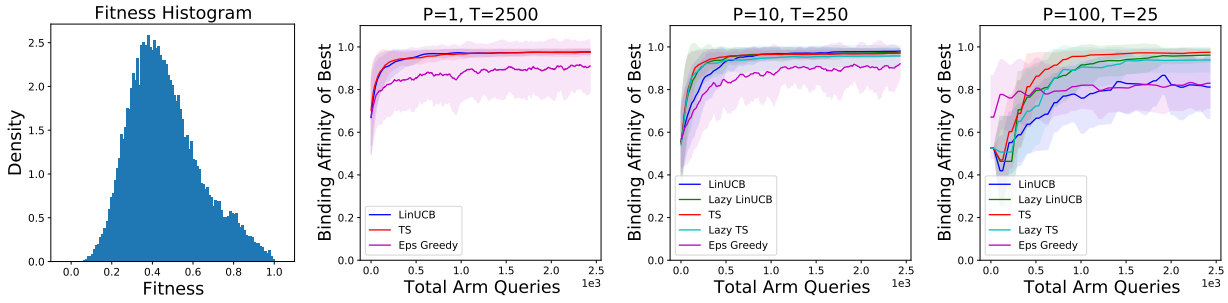

Figure 4.6: TFBinding best arm with linear features. Leftmost: The fitness distribution of the dataset. From left to right: The best smoothed binding affinity for each round with error bars indicating standard deviation with $P = 1, 10,$ and 100, respectively.

4.6 Conclusion

In this work, we present several parallel (contextual) linear bandit algorithms inspired by the optimism principle. Utilizing the notion of covariance stability, we provide a unified analysis of their regret in a variety of settings. Our regret upper bounds establish the performance of these algorithms is nearly identical to their sequential counterparts (with the same total number of arm queries) up to a burn-in term which may depend on the context set geometry. Finally, we show that the parallelism gains suggested by our theory can also be achieved in several real datasets motivated by practical design problems and demonstrate the importance of diversity in problems that contain model misspecification. Interesting directions for future work including extending the results herein to a suitably defined notion of best-arm identification. Similarly, understanding the impact of parallelism in simple, greedy heuristic algorithms (which nonetheless perform well in practice [12, 83]) is another important direction. Another interesting direction is leveraging parallel processors to devise online model selection and hyperparameter tuning strategies. Lastly, in applications such as protein engineering, it is often of interest to discover a diverse set of high-reward sequences under suitable notions of diversity.

4.7 Additional Experimental Details

In this section we provide all of the experimental details for training the bandit algorithms.

Hyperparameters

The synthetic hyperparameters were fixed to the theoretical values. For the randomly initialized neural network experiments, the grid for the regularizer λ, norm bound S, and noise subgaussianity R were: $\{0.01, 0.1, 1.0, 10.0, 100.0\}$ which was selected post-hoc for each experiment. For the superconductor experiments, the grid for all 3 parameters were: $\{0.1, 1.0, 10.0\}$. For the transcription factor binding dataset, the parameters grid was $\lambda = \{1.0, 10.0\}$, $R = \{0.01, 0.1, 1.0, 10.0, 100.0, 1000.0\}$, and $S = \{0.01, 0.1, 1.0, 10.0, 100.0, 1000.0\}$. For ϵ-greedy, the parameter grid was set over $\epsilon = \{0.01, 0.02, \ldots, 0.99\}$ across all relevant experiments.

Feature Engineering for Random Neural Network

Two feature sets were considered:

- Linear features x_i was encoded as a 14-length feature vector
- Linear + Quadratic features where interaction terms of x_i were included in 105 quadratic features alongside the original 14 linear features into a 119 features.

The resulting parallel regret plots are shown in Fig. 4.8.

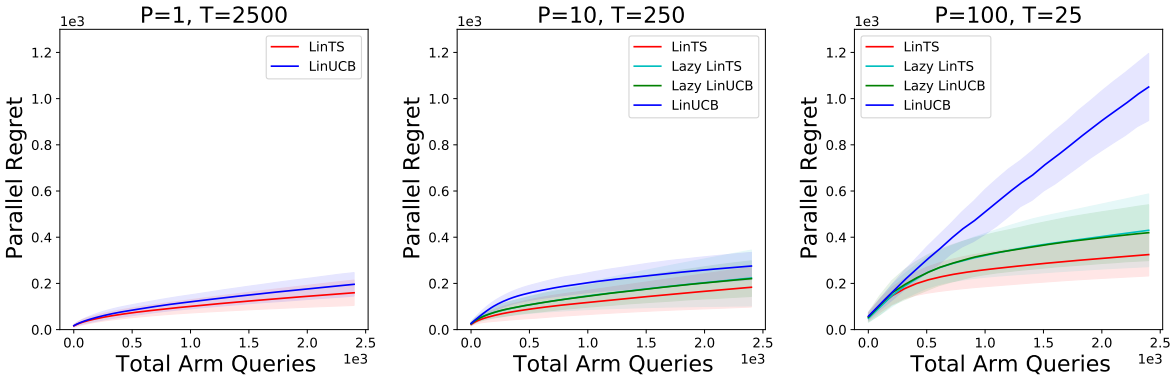

Figure 4.7: TFBinding parallel regret with linear features. From left to right: $P = 1$, $P = 10$, and $P = 100$.

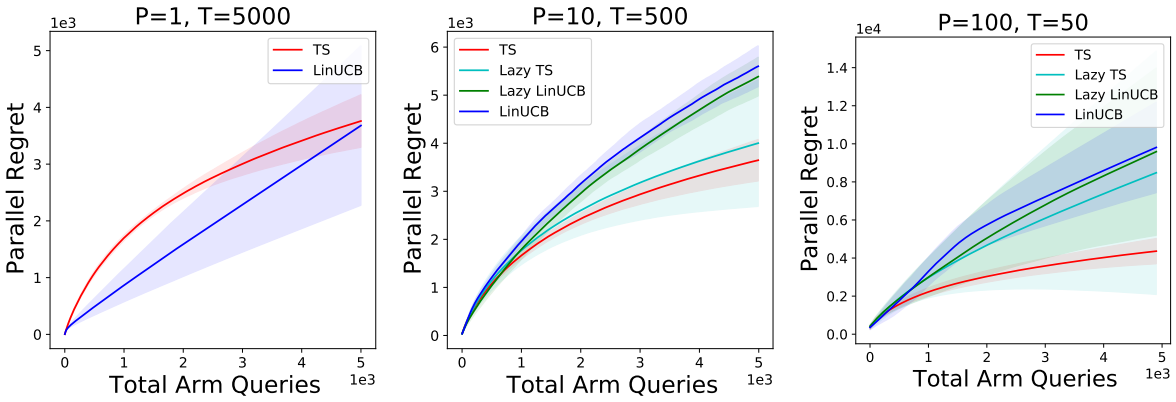

Figure 4.8: RandomNN with Linear features. From left to right: $P = 1$, $P = 10$, and $P = 100$.

Feature Engineering for Transcription Factor Binding

Three feature sets were considered:

- Linear features $\mathbf{x}_i$ was one-hot encoded into a 32-length feature vector.
- 250 random ReLU features where a 250×32 random matrix $\mathbf{W}$ is sampled such that $\mathbf{W}_{ij} \sim \text{Normal}(0, 1)$. Then, the feature map was evaluated as:

$$\phi(\mathbf{x}_i) = \frac{1}{\sqrt{250}} \text{ReLU}\left(\frac{\mathbf{W}\mathbf{x}_i}{\sqrt{32}}\right)$$

- Linear + Quadratic features where $\mathbf{x}_i$ was one-hot encode and all 32 linear features and 528 quadratic features were combined into a 560-length vector.

The off-line test R^2 (without added noise in the y_i) on a train set of $TP = 2500$ matching the number of total arm queries yields values $0.15, 0.26, 0.29$ for linear, ReLU, and quadratic respectively. This matches with the number of features and level of expressivity of the model class. However, as shown in Figs. 4.9 and 4.10 the linear features perform the best followed by the ReLU features, and then the quadratic features. One can gain further insight by examining the off-line test R^2 for a smaller training size and see that the larger feature expansions accrue variance making the linear features perform the best. This matches our understanding that model fitting is less statistically efficient in an online setting.

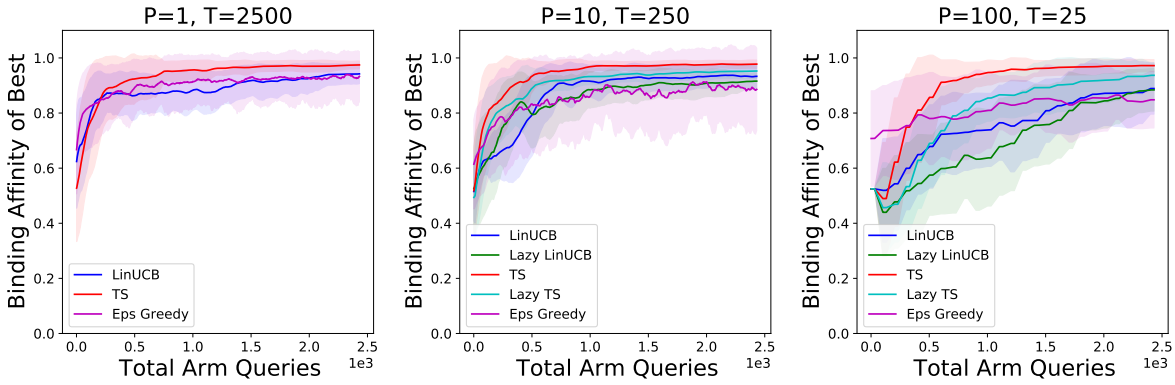

Figure 4.9: TFBinding best arm with ReLU features. From left to right: $P = 1$, $P = 10$, and $P = 100$.

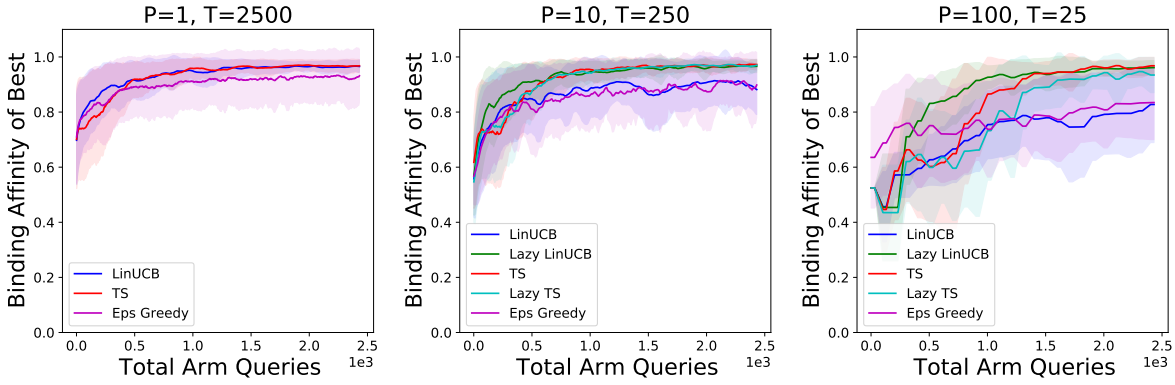

Figure 4.10: TFBinding best arm with quadratic features. From left to right: $P = 1$, $P = 10$, and $P = 100$.

4.8 Proofs

In this section we provide the proofs of the regret upper bounds and lower bound for all of the algorithms considered.

Proofs for Section 4.2

Here we include the Proof of Theorem 2.

Proof of Theorem 2. We first decompose the regret for Algorithm 1 by splitting into the linear term and misspecification component.

$$\begin{aligned}\mathcal{R}(T,P) &= \sum_{t=1}^{T}\left(\sum_{p=1}^{P} f(\mathbf{x}_{t,p}^{\star}) - f(\mathbf{x}_{t,p})\right) \\ &= \sum_{t=1}^{T}\left(\sum_{p=1}^{P}\langle \mathbf{x}_{t,p}^{\star} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^{\star}\rangle\right) + \sum_{t=1}^{T}\Big(\sum_{p=1}^{P} f(\mathbf{x}_{t,p}^{\star}) - \langle \mathbf{x}_{t,p}^{\star}, \boldsymbol{\theta}^{\star}\rangle + f(\mathbf{x}_{t,p}) - \langle \mathbf{x}_{t,p}, \boldsymbol{\theta}^{\star}\rangle\Big).\end{aligned}$$

The second term can be immediately upper bounded as,

$$\sum_{t=1}^{T}\Big(\sum_{p=1}^{P} f(\mathbf{x}_{t,p}^{\star}) - \langle \mathbf{x}_{t,p}^{\star}, \boldsymbol{\theta}^{\star}\rangle + f(\mathbf{x}_{t,p}) - \langle \mathbf{x}_{t,p}, \boldsymbol{\theta}^{\star}\rangle\Big) \leq 2\epsilon TP.$$

using Assumption 3. We now approach the linearized reward term. We split this term in each round over the event $\mathcal{D}_t$,

$$\begin{aligned}\sum_{t=1}^{T}\left(\sum_{p=1}^{P}\langle \mathbf{x}_{t,p}^{\star} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^{\star}\rangle\right) &= \sum_{t=1}^{T}\mathbb{1}[\mathcal{D}_t]\left(\sum_{p=1}^{P}\langle \mathbf{x}_{t,p}^{\star} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^{\star}\rangle\right) \\ &\quad + \sum_{t=1}^{T}\mathbb{1}[\mathcal{D}_t^{c}]\left(\sum_{p=1}^{P}\langle \mathbf{x}_{t,p}^{\star} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^{\star}\rangle\right)\end{aligned}$$

The first term here can be bounded using Assumptions 2 and 3 along with the Cauchy-Schwarz inequality which gives $\left(\sum_{p=1}^{P}\langle \mathbf{x}_{t,p}^{\star} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^{\star}\rangle\right) \leq 2LSP$ so:

$$\sum_{t=1}^{T}\mathbb{1}[\mathcal{D}_t]\left(\sum_{p=1}^{P}\langle \mathbf{x}_{t,p}^{\star} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^{\star}\rangle\right) \leq 2LSP\sum_{t=1}^{T}\mathbb{1}[\mathcal{D}_t].$$

Note the above bounds hold for any choices of $\mathbf{x}_{t,p} \in \mathcal{X}_{t,p}$ selected by any doubling-round routine. We now turn our attention to the second term. For this term we use essentially the same techniques to bound the instantaneous regret by the exact same value for both Algorithm 1 and Algorithm 2, but separate the analysis into two cases for clarity.

- For Algorithm 1 we refer to the optimistic model of processor p at round t as:

$$\tilde{\boldsymbol{\theta}}_{t,p} = \underset{\theta \in \mathcal{C}_{t,1}(\hat{\boldsymbol{\theta}}_t, \mathbf{V}_{t,1}, \beta_t(\delta), \epsilon)}{\operatorname{argmax}} \left(\max_{\mathbf{x} \in \mathcal{X}_{t,p}} \langle \mathbf{x}, \boldsymbol{\theta} \rangle \right)$$

for Algorithm 1. Conditioned on the event in Theorem 9–which we denote $\mathcal{E}_1$–the models $\tilde{\boldsymbol{\theta}}_{t,p}$ are optimistic:

$$\langle \mathbf{x}_{t,p}, \tilde{\boldsymbol{\theta}}_{t,p} \rangle \mathbb{1}[\mathcal{E}_1] \geq \langle \mathbf{x}^\star_{t,p}, \boldsymbol{\theta}^\star \rangle \mathbb{1}[\mathcal{E}_1].$$

Hence,

$$\mathbb{1}[\mathcal{E}_1]\mathbb{1}[\mathcal{D}_t^c]\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star \rangle \leq \mathbb{1}[\mathcal{E}_1]\mathbb{1}[\mathcal{D}_t^c]\langle \mathbf{x}_{t,p}, \tilde{\boldsymbol{\theta}}_{t,p} - \boldsymbol{\theta}^\star \rangle \leq$$
$$\mathbb{1}[\mathcal{E}_1]\mathbb{1}[\mathcal{D}_t^c]\|\mathbf{x}_{t,p}\|_{\mathbf{V}^{-1}_{t,p}}\|\tilde{\boldsymbol{\theta}}_{t,p} - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,p}} \leq 2\sqrt{2}\mathbb{1}[\mathcal{E}]\mathbb{1}[\mathcal{D}_t^c]\|\mathbf{x}_{t,p}\|_{\mathbf{V}^{-1}_{t,p}}\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,1}}$$

using optimism in the first inequality, Cauchy-Schwartz in the second, and the fact that on event $\mathcal{D}_t^c$ round t is not a generalized doubling round in the in the final inequality. Finally, recall on the event $\mathcal{E}_1$, $\mathbb{1}[\mathcal{E}_1]\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,1}} \leq (\sqrt{\beta_t(\delta)} + \sqrt{(t-1)P\epsilon}))\mathbb{1}[\mathcal{E}_1]$ and note $\beta_t(\delta)$ is an increasing function of t so $\beta_t(\delta) \leq \beta_T(\delta)$ for all $t \leq T$. Hence it follows,

$$\mathbb{1}[\mathcal{E}_1]\mathbb{1}[\mathcal{D}_t^c]\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star \rangle \leq \mathbb{1}[\mathcal{E}_1]2\sqrt{2}(\sqrt{\beta_T(\delta)} + \sqrt{(t-1)P\epsilon})\|\mathbf{x}_{t,p}\|_{\mathbf{V}^{-1}_{t,p}}$$

additionally relaxing $\mathbb{1}[\mathcal{D}_t^c] \leq 1$.

- For Algorithm 2 we also refer to the optimistic model of processor p at round t as:

$$\tilde{\boldsymbol{\theta}}'_{t,p} = \underset{\theta \in \mathcal{C}_{t,p}(\hat{\boldsymbol{\theta}}_t, \mathbf{V}_{t,p}, 2\beta_t(\delta), 2\epsilon)}{\operatorname{argmax}} \left(\max_{\mathbf{x} \in \mathcal{X}_{t,p}} \langle \mathbf{x}, \boldsymbol{\theta} \rangle \right).$$

Conditioned on the event of Theorem 10 restricted to not being a doubling round, the models $\tilde{\boldsymbol{\theta}}'_{t,p}$ are optimistic:

$$\langle \mathbf{x}_{t,p}, \tilde{\boldsymbol{\theta}}'_{t,p} \rangle \mathbb{1}[\mathcal{E}_2]\mathbb{1}[\mathcal{D}_t^c] \geq \langle \mathbf{x}^\star_{t,p}, \boldsymbol{\theta}^\star \rangle \mathbb{1}[\mathcal{E}_2]\mathbb{1}[\mathcal{D}_t^c].$$

Hence,

$$\mathbb{1}[\mathcal{E}_2]\mathbb{1}[\mathcal{D}_t^c]\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star \rangle \leq \mathbb{1}[\mathcal{E}_2]\mathbb{1}[\mathcal{D}_t^c]\langle \mathbf{x}_{t,p}, \tilde{\boldsymbol{\theta}}'_{t,p} - \boldsymbol{\theta}^\star \rangle \leq$$
$$\mathbb{1}[\mathcal{E}_2]\mathbb{1}[\mathcal{D}_t^c]\|\mathbf{x}_{t,p}\|_{\mathbf{V}^{-1}_{t,p}}\|\tilde{\boldsymbol{\theta}}'_{t,p} - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,p}}$$

using optimism in the first inequality, Cauchy-Schwartz in the second. Finally, on the event $\mathcal{E}_2 \cap \mathcal{D}_t^c$, $\mathbb{1}[\mathcal{E}_2]\mathbb{1}[\mathcal{D}_t^c]\|\tilde{\boldsymbol{\theta}}'_{t,p} - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,p}} \leq 2\sqrt{2}\sqrt{\beta_t(\delta)}\mathbb{1}[\mathcal{E}_2]\mathbb{1}[\mathcal{D}_t^c] \leq 2\sqrt{2}\sqrt{\beta_T(\delta)}\mathbb{1}[\mathcal{E}_2]\mathbb{1}[\mathcal{D}_t^c]$. Hence it follows,

$$\mathbb{1}[\mathcal{E}_2]\mathbb{1}[\mathcal{D}_t^c]\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star \rangle \leq 2\sqrt{2}\mathbb{1}[\mathcal{E}_2](\sqrt{\beta_T(\delta)} + \sqrt{(t-1)P\epsilon})\|\mathbf{x}_{t,p}\|_{\mathbf{V}^{-1}_{t,p}}$$

additionally relaxing $\mathbb{1}[\mathcal{D}_t^c] \leq 1$.

The remainder of the proof follows identically for both Algorithms 1 and 2. Without loss of generality we use $\mathcal{E}$ to refer to either event $\mathcal{E}_1$ or $\mathcal{E}_2$ in the following (note both hold with probability at least $1-\delta$ by Theorems 7 and 8). Recalling that the instantaneous regret is $\leq 2LS$ we can combine this bound with the aforementioned bounds to conclude that,

$$\begin{aligned}
&\mathbb{1}[\mathcal{E}] \cdot \left(\sum_{t=1}^{T} \mathbb{1}[\mathcal{D}_t^c] \left(\sum_{p=1}^{P} \langle \mathbf{x}_{t,p}^\star - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star \rangle \right) \right) \\
&\overset{(i)}{\leq} \mathbb{1}[\mathcal{E}] \cdot \sqrt{TP \left(\sum_{t=1}^{T} \mathbb{1}[\mathcal{D}_t^c] \left(\sum_{p=1}^{P} \langle \mathbf{x}_{t,p}^\star - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star \rangle^2 \right) \right)} \\
&\overset{(ii)}{\leq} 4\sqrt{2}\mathbb{1}[\mathcal{E}] \sqrt{TP \sum_{t=1}^{T} \sum_{p=1}^{P} \min((LS)^2, 2(\beta_T(\delta) + (t-1)P\epsilon^2) \|\mathbf{x}_{t,p}\|^2_{\mathbf{V}_{t,p}^{-1}})} \\
&\overset{(iii)}{\leq} 4\sqrt{2}\mathbb{1}[\mathcal{E}] \sqrt{TP \sum_{t=1}^{T} \sum_{p=1}^{P} \min((LS)^2, 2(\beta_T(\delta) + TP\epsilon^2) \|\mathbf{x}_{t,p}\|^2_{\mathbf{V}_{t,p}^{-1}})} \\
&\overset{(iv)}{\leq} 4\sqrt{2}\mathbb{1}[\mathcal{E}] \sqrt{TP2(\beta_T(\delta) + TP\epsilon^2) \max\left(2, \frac{(LS)^2}{2(\beta_T(\delta) + TP\epsilon^2)}\right)} \\
&\quad \cdot \sqrt{\sum_{t=1}^{T} \sum_{p=1}^{P} \log\left(1 + \sum_{t=1}^{T} \sum_{p=1}^{P} \|\mathbf{x}_{t,p}\|^2_{\mathbf{V}_{t,p}^{-1}}\right)} \\
&\overset{(v)}{\leq} 8\mathbb{1}[\mathcal{E}] \sqrt{TP} \max(\sqrt{2}\sqrt{\beta_T(\delta)} + \sqrt{TP}\epsilon, LS) \cdot \sqrt{d \log\left(1 + \frac{TPL^2}{\lambda}\right)}.
\end{aligned}$$

Inequality (i) follow by Cauchy-Schwarz. Inequality (ii) employs both our bounds on the instantaneous regret. Inequality (iii) follows by upper bounding $t-1 \leq T$ for the misspecification term. Inequality (iv) follows because for all $a, x > 0$, we have $\min(a, x) \leq \max(2, a)\log(1+x)$. Inequality (v) follows because

$$\sum_{t=1}^{T} \sum_{p=1}^{P} \log(1 + \|\mathbf{x}_{t,p}\|^2_{\mathbf{V}_{t,p}^{-1}}) \leq d \log\left(1 + \frac{PTL^2}{\lambda})\right)$$

by instantiating Lemma 11. Assembling the bounds in the original regret splitting over doubling rounds and accounting for the original misspecification term shows that,

$$\begin{aligned}
\mathbb{1}[\mathcal{E}]\mathcal{R}(T,P) \leq \mathbb{1}[\mathcal{E}]8 \cdot \Bigg(&LSP \sum_{t=1}^{T} \mathbb{1}[\mathcal{D}_t] + \sqrt{TP} \max(\sqrt{2}(\sqrt{\beta_T(\delta)} + \sqrt{TP}\epsilon), LS) \\
&\cdot \sqrt{d \log\left(1 + \frac{TPL^2}{\lambda}\right)} + \epsilon TP \Bigg)
\end{aligned}$$

where $\sqrt{\beta_T(\delta)} \leq R\sqrt{d \log\left(\frac{1+TPL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}S$. This inequality holds on the event $\mathcal{E}$ which occurs with probability at least $1-\delta$ for both Algorithm 1 and Algorithm 2. Inserting this value for $\beta_T(\delta)$ and hiding logarithmic factors shows on the event $\mathcal{E}$,

$$\mathcal{R}(T,P) \leq \tilde{O}\left(LSP \cdot \sum_{t=1}^{T} \mathbb{1}[\mathcal{D}_t] + \sqrt{dTP}\max(\sqrt{2}(R\sqrt{d} + \sqrt{\lambda}S + \sqrt{TP}\epsilon), LS)\right)$$

□

Proofs of Section 4.2

We start by stating a folklore lemma regarding the anti-concentration properties of a Gaussian distribution.

Concentration and Anti-Concentration properties of the Gaussian distribution

Lemma 4. *Let X be a random variable distributed according to $\mathcal{N}(\mu, \sigma^2)$, a one dimensional Gaussian distribution with mean μ and variance σ^2. The following holds:*

$$\mathbb{P}\left(X - \mu \geq \tau\right) \geq \frac{1}{\sqrt{2\pi}} \frac{\sigma\tau}{\tau^2 + \sigma^2} \exp\left(-\frac{\tau^2}{2\sigma^2}\right)$$

We will also make use of the following concentration inequality for Lipschitz functions of Gaussian vectors:

Theorem 5 (Theorem 2.4 in [93])**.** *Let $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d)$ be a standard Gaussian vector and let $f : \mathbb{R}^d \to \mathbb{R}$ be $L-$Lipschitz with respect to the Euclidean norm. Then the variable $f(\boldsymbol{\eta}) - \mathbb{E}\left[f(\boldsymbol{\eta})\right]$ is subgaussian with parameter at most L and hence:*

$$\mathbb{P}\left(f(X) \geq \mathbb{E}\left[f(X)\right] + t\right) \leq \exp\left(-\frac{t^2}{2L^2}\right)$$

We'll make use of these two results to prove Lemma 5 which we restate for the reader's convencience:

Lemma 5. *The Gaussian distribution satisfies (anticoncentration) for every $\mathbf{v} \in \mathbb{R}^d$ with $\|\mathbf{v}\| = 1$:*

$$\mathbb{P}_{\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d)}\left(\mathbf{v}^\top \boldsymbol{\eta} \geq 1\right) \geq \frac{1}{4}. \tag{4.17}$$

And (concentration), $\forall \delta \in (0,1)$:

$$\mathbb{P}_{\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d)}\left(\|\boldsymbol{\eta}\| \leq \sqrt{d} + \sqrt{2\log\left(\frac{1}{\delta}\right)}\right) \geq 1 - \delta. \tag{4.18}$$

Proof. Equation 4.17 is a simple consequence of the following two observations:

1. For any unit norm vector $\mathbf{v} \in \mathbb{R}^d$ the random variable $X = \mathbf{v}^\top \eta$ is distributed as a one dimensional Gaussian with unit variance $\mathcal{N}(0, 1)$.
2. Setting parameters $\mu = 0, \sigma = 1$, and $\tau = 1$ Lemma 4 implies that $\mathbb{P}(X \geq 1) \geq \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{2} \exp(-\frac{1}{2}) \geq$.

Equation 4.18 instead follows from Theorem 5. Since the function $f(\cdot) = \|\cdot\|$ is $1-$Lipschitz and $\mathbb{E}_{\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d)} [\|\boldsymbol{\eta}\|] \leq (\mathbb{E}_{\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d)} [\|\boldsymbol{\eta}\|^2])^{1/2} = \sqrt{d}$:

$$\mathbb{P}_{\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d)} \left(\|\boldsymbol{\eta}\| \geq \sqrt{d} + \sqrt{2 \log\left(\frac{1}{\delta}\right)} \right) \leq \delta.$$

The result follows. □

Recall that:

$$\sqrt{\beta_t(\delta)} = R\sqrt{\log\left(\frac{\det(\mathbf{V}_{t-1,0})}{\lambda^d \delta^2}\right)} + \sqrt{\lambda} S \leq R\sqrt{d \log\left(\frac{1 + tPL^2/\lambda}{\delta}\right)} + \sqrt{\lambda} S$$

Concentration of $\tilde{\boldsymbol{\theta}}_{t,p}$

The main objective of this section is to show that with high probability the sampled parameter $\tilde{\boldsymbol{\theta}}_{t,p}$ is not too far from the true parameter $\boldsymbol{\theta}_\star$ for all times t and processors p. The result is encapsuled by Lemma 6.

Lemma 6. *Let:*

$$\gamma_{t,p}(\delta) := \sqrt{2}(\sqrt{\beta_t(\delta)} + \sqrt{(t-1)P\epsilon}) \left(\sqrt{d} + 2\sqrt{\log\left(\frac{t(P-1)+p}{\delta}\right)} + 1 \right).$$

The following conditional probability bound holds:

$$\mathbb{P}\left(\mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}] \|\tilde{\boldsymbol{\theta}}_{t,p} - \boldsymbol{\theta}_\star\|_{\mathbf{V}_{t,p}} \geq \gamma_{t,p}(\delta) \middle| \mathcal{F}_{t,p-1} \right) \leq \frac{\delta}{2(t(P-1)+p)^2} \tag{4.19}$$

And therefore with probability at least $1 - 2\delta$ and for all $t \in N$ simultaneously:

$$\|\tilde{\boldsymbol{\theta}}_{t,p} - \boldsymbol{\theta}_\star\|_{\mathbf{V}_{t,p}} \leq \gamma_{t,p}(\delta) \tag{4.20}$$

We refer to this event as $\mathcal{E}'$.

In order to prove Lemma 6 let's start by showing that for any time-step t and processor t the sample $\tilde{\boldsymbol{\theta}}_{t-1,p}$ is close to $\boldsymbol{\theta}_\star$ with high probability:

Lemma 7. *The following conditional probability bound holds:*

$$\mathbb{P}\left(\|\tilde{\boldsymbol{\theta}}_{t,p}-\hat{\boldsymbol{\theta}}_t\|_{\mathbf{V}_{t,p}} \geq \sqrt{2\beta_t(\delta)}\left(\sqrt{d}+2\sqrt{\log\left(\frac{t(P-1)+p}{\delta}\right)}\right)\Big|\mathcal{F}_{t,p-1}\right) \leq \frac{\delta}{2(t(P-1)+p)^2}$$

Where $\mathcal{F}_{t,p-1}$ corresponds to the sigma algebra generated by all the events up to and including the reveal of contexts $\mathcal{X}_{t,p}$. And therefore with probability at least $1-\delta$ simultaneously and unconditionally for all $t \in \mathbb{N}$:

$$\|\tilde{\boldsymbol{\theta}}_{t,p}-\hat{\boldsymbol{\theta}}_t\|_{\mathbf{V}_{t,p}} \leq \sqrt{2}(\sqrt{\beta_t(\delta)}+\sqrt{(t-1)P\epsilon})\left(\sqrt{d}+2\sqrt{\log\left(\frac{t(P-1)+p}{\delta}\right)}\right)$$

Proof. In order to bound $\|\tilde{\boldsymbol{\theta}}_{t,p}-\hat{\boldsymbol{\theta}}_t\|_{\mathbf{V}_{t,p}}$ we make use of Lemma 5. Observe that by definition:

$$\|\tilde{\boldsymbol{\theta}}_{t,p}-\hat{\boldsymbol{\theta}}_t\|_{\mathbf{V}_{t,p}} = \sqrt{2}(\sqrt{\beta_t(\delta)}+\sqrt{(t-1)P\epsilon})\|\boldsymbol{\eta}_{t,p}\|_2. \tag{4.21}$$

Therefore a simple use of Lemma 5 implies that (concentration):

$$\mathbb{P}_{\boldsymbol{\eta}_{t,p}\sim\mathcal{N}(\mathbf{0},\mathbb{I}_d)}\left(\|\boldsymbol{\eta}_{t,p}\| \leq \sqrt{d}+2\sqrt{\log\left(\frac{t(P-1)+p}{\delta}\right)}\Big|\mathcal{F}_{t,p-1}\right) \leq \frac{\delta}{2(t(P-1)+p)^2}.$$

And therefore as a consequence of Equation 4.21:

$$\mathbb{P}\left(\|\tilde{\boldsymbol{\theta}}_{t,p}-\hat{\boldsymbol{\theta}}_t\|_{\mathbf{V}_{t,p}} \geq \sqrt{2}(\sqrt{\beta_t(\delta)}+\sqrt{(t-1)P\epsilon})\left(\sqrt{d}+2\sqrt{\log\left(\frac{t(P-1)+p}{\delta}\right)}\right)\Big|\mathcal{F}_{t,p-1}\right)$$
$$\leq \frac{\delta}{2(t(P-1)+p)^2}$$

Furthermore, a simple union bound implies that for all $t \in \mathbb{N}$:

$$\mathbb{P}\left(\exists t \text{ s.t. } \|\boldsymbol{\eta}_{t,p}\|_2 \geq \sqrt{d}+2\sqrt{\log\left(\frac{t(P-1)+p}{\delta}\right)}\right) \leq \frac{\delta}{2}\sum_{t=1}^{\infty}\sum_{p=1}^{P}\frac{1}{(t(P-1)+p)^2} \leq \delta \tag{4.22}$$

Combining equations 4.21 and 4.22 yields:

$$\mathbb{P}\left(\|\tilde{\boldsymbol{\theta}}_{t,p}-\hat{\boldsymbol{\theta}}_t\|_{\mathbf{V}_{t,p}} \leq \sqrt{2}(\sqrt{\beta_t(\delta)}+\sqrt{(t-1)P\epsilon})\left(\sqrt{d}+2\sqrt{\log\left(\frac{t(P-1)+p}{\delta}\right)}\right)\right) \leq \delta$$

□

Lemma 7, conditioning on the $1-\delta$ probability event $\mathcal{E}$, a simple use of the triangle inequality along with the identity $\sum_{p=1}^{P}\sum_{t=1}^{\infty}\frac{1}{(t(P-1)+p)^2} = \frac{\pi^2}{6} < 2$ finalizes the proof of Lemma 6. From now on we will denote as $\mathcal{E}'$ to the $1-2\delta$ probability event defined by Lemma 6.

Anti-concentration of $\tilde{\theta}_{t,p}$

The main objective of this section will be to prove the following upper bound for the instantaneous regret $\boldsymbol{\theta}_\star^\top \mathbf{x}_{t,p}^\star - \boldsymbol{\theta}_\star^\top \mathbf{x}_{t,p}$ in the event the round is not a doubling round and $\mathcal{E}'$ holds. This is one of the main components of the proof of Theorem 3.

Lemma 8. *Let $\tilde{\boldsymbol{\theta}}'_{t,p}$ be a copy of $\tilde{\boldsymbol{\theta}}_{t,p}$, equally distributed to $\tilde{\boldsymbol{\theta}}_{t,p}$ and independent of it conditionally on $\mathcal{F}_{t,p-1}$. We call $\mathbf{x}'_{t,p}$ to the resulting argmax action $\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}_{t,p}}\langle\tilde{\boldsymbol{\theta}}'_{t,p}, \mathbf{x}\rangle$. The following inequality holds:*

$$\mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\left(\boldsymbol{\theta}_\star^\top \mathbf{x}_{t,p}^\star - \boldsymbol{\theta}_\star^\top \mathbf{x}_{t,p}\right) \leq \frac{2\gamma_{t,p}(\delta)}{\frac{1}{4} - \frac{\delta}{2(t(P-1)+p)^2}} \mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\mathbb{E}\left[\|\mathbf{x}'_{t,p}\|_{\mathbf{V}_{t,p}^{-1}} \middle| \mathcal{F}_{t,p-1}\right] + \gamma_{t,p}(\delta)\mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\|\mathbf{x}_{t,p}\|_{\mathbf{V}_{t,p}^{-1}}.$$

Before proving Lemma 8 we show that with constant probability, the estimated value of the action taken at time and processor tuple (t,p) is optimistic with constant probability:

Lemma 9. *For all $t \in N$:*

$$\mathbb{P}\left(\tilde{\boldsymbol{\theta}}_{t,p}^\top \mathbf{x}_{t,p} \geq \boldsymbol{\theta}_\star^\top \mathbf{x}_{t,p}^\star \middle| \mathcal{F}_{t,p-1}, \mathcal{E}\right) \geq \frac{1}{4}$$

Proof. Recall that whenever $\mathcal{E}$ holds:

$$\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star\|_{\mathbf{V}_{t,p}} \leq \sqrt{2}(\sqrt{\beta_t(\delta)} + \sqrt{(t-1)P}\epsilon).$$

Notice that by definition $\mathbf{x}_{t,p}$ satisfies:

$$\tilde{\boldsymbol{\theta}}_{t,p}^\top \mathbf{x}_{t,p} \geq \tilde{\boldsymbol{\theta}}_{t,p}^\top \mathbf{x}_{t,p}^\star.$$

Therefore:

$$\begin{aligned}
\tilde{\boldsymbol{\theta}}_{t,p}^\top \mathbf{x}_{t,p}^\star &= \left(\tilde{\boldsymbol{\theta}}_{t,p} - \hat{\boldsymbol{\theta}}_t\right)^\top \mathbf{x}_{t,p}^\star + \left(\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star\right)^\top \mathbf{x}_{t,p}^\star + \boldsymbol{\theta}_\star^\top \mathbf{x}_{t,p}^\star \\
&\overset{(i)}{\geq} \sqrt{2}(\sqrt{\beta_t(\delta)} + \sqrt{(t-1)P}\epsilon)\boldsymbol{\eta}_{t,p}^\top \mathbf{V}_{t,p}^{-1/2}\mathbf{x}_{t,p}^\star - \|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star\|_{\mathbf{V}_{t,p}}\|\mathbf{x}_{t,p}^\star\|_{\mathbf{V}_{t,p}^{-1}} + \boldsymbol{\theta}_\star^\top \mathbf{x}_{t,p}^\star \\
&\overset{(ii)}{\geq} \sqrt{2}(\sqrt{\beta_t(\delta)} + \sqrt{(t-1)P}\epsilon)\boldsymbol{\eta}_{t,p}^\top \mathbf{V}_{t,p}^{-1/2}\mathbf{x}_{t,p}^\star \\
&\quad - \sqrt{2}(\sqrt{\beta_t(\delta)} + \sqrt{(t-1)P}\epsilon)\|\mathbf{x}_{t,p}^\star\|_{\mathbf{V}_{t,p}^{-1}} + \boldsymbol{\theta}_\star^\top \mathbf{x}_{t,p}^\star \\
&= \sqrt{2}(\sqrt{\beta_t(\delta)} + \sqrt{(t-1)P}\epsilon)\left(\boldsymbol{\eta}_{t,p}^\top \mathbf{V}_{t,p}^{-1/2}\mathbf{x}_{t,p}^\star - \|\mathbf{V}_{t,p}^{-1/2}\mathbf{x}_{t,p}^\star\|_2\right).
\end{aligned}$$

Inequality i holds as a consequence of Cauchy Schwartz inequality. Inequality (ii) holds by conditioning on $\mathcal{E}$ and because $t \in N$.

By Equation 4.17 in Lemma 5, and by noting that $\mathbf{x}_{t,p}^\star$ is conditionally independent of $\boldsymbol{\eta}_{t,p}$, we can infer that $\boldsymbol{\eta}_{t,p}^\top \mathbf{V}_{t,p}^{-1/2}\mathbf{x}_{t,p}^\star \geq \|\mathbf{V}_{t,p}^{-1/2}\mathbf{x}_{t,p}^\star\|_2$ with probability at least $1/4$. The result follows.

□

Let's define the set of optimistic model parameters:

$$\boldsymbol{\Theta}_{t,p} = \{\boldsymbol{\theta} \in \mathbb{R}^d \text{ s.t. } \max_{\mathbf{x}\in\mathcal{X}_{t,p}} \mathbf{x}^\top\boldsymbol{\theta} \geq (\mathbf{x}^\star_{t,p})^\top\boldsymbol{\theta}_\star\}.$$

Where $\mathcal{D}_t$ denotes the event that round t is a doubling round. We now show how to bound the instantaneous regret $r_{t,p}$ during all rounds by using these results:

Proof of Lemma 8. Recall that $\mathcal{E}'$ is the event defined by Equation 4.20 in Lemma 6 applied to the $\{\tilde{\boldsymbol{\theta}}_{t,p}\}_{t,p}$ sequence. Define $\{\mathcal{E}''_{t,p}\}_{t,p}$ be the corresponding event family defined by Equation 4.19 in Lemma 6 applied to the $\{\tilde{\boldsymbol{\theta}}'_{t,p}\}_{t,p}$ sequence. It follows that $\mathbb{P}(\mathcal{E}''_{t,p}|\mathcal{F}_{t,p-1}) \geq 1 - \frac{\delta}{2(t(P-1)+p)^2}$. Notice that if $\mathcal{D}_t^c \cap \mathcal{E}'$ holds (meaning $\|\tilde{\boldsymbol{\theta}}_{t,p} - \boldsymbol{\theta}_\star\|_{\mathbf{V}_{t,p}} \leq \delta_{t,p}$ and because $\mathbf{x}_{t,p} = \text{argmax}_{\mathbf{x}\in\mathcal{X}_{t,p}} \tilde{\boldsymbol{\theta}}_{t,p}^\top\mathbf{x}_{t,p}$ then:

$$\tilde{\boldsymbol{\theta}}_{t,p}^\top\mathbf{x}_{t,p} \geq \inf_{\boldsymbol{\theta}\in\mathcal{C}(\boldsymbol{\theta}_\star,\mathbf{V}_{t,p},\gamma_{t,p}(\delta))} \max_{\mathbf{x}\in\mathcal{X}_{t,p}} \boldsymbol{\theta}_{t,p}^\top\mathbf{x} := \bar{\boldsymbol{\theta}}_{t,p}^\top\bar{\mathbf{x}}_{t,p}.$$

When $\tilde{\boldsymbol{\theta}}'_{t,p}$ is optimistic:

$$\boldsymbol{\theta}_\star^\top\mathbf{x}^\star_{t,p} - \tilde{\boldsymbol{\theta}}_{t,p}^\top\mathbf{x}_{t,p} \leq \langle\tilde{\boldsymbol{\theta}}'_{t,p}, \mathbf{x}'_{t,p}\rangle - \bar{\boldsymbol{\theta}}_{t,p}^\top\bar{\mathbf{x}}_{t,p} \quad \Big|\tilde{\boldsymbol{\theta}}'_{t,p} \in \boldsymbol{\Theta}_{t,p}. \tag{4.23}$$

Therefore:

$$\begin{aligned}
\mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\left(\boldsymbol{\theta}_\star^\top\mathbf{x}^\star_{t,p} - \boldsymbol{\theta}_\star^\top\mathbf{x}_{t,p}\right) &= \mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\left(\boldsymbol{\theta}_\star^\top\mathbf{x}^\star_{t,p} - \tilde{\boldsymbol{\theta}}_{t,p}^\top\mathbf{x}_{t,p}\right) + \mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\left(\tilde{\boldsymbol{\theta}}_{t,p}^\top\mathbf{x}_{t,p} - \boldsymbol{\theta}_\star^\top\mathbf{x}_{t,p}\right) \\
&\leq \mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\left(\boldsymbol{\theta}_\star^\top\mathbf{x}^\star_{t,p} - \tilde{\boldsymbol{\theta}}_{t,p}^\top\mathbf{x}_{t,p}\right) + \mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\|\tilde{\boldsymbol{\theta}}_{t,p} \\
&\quad - \boldsymbol{\theta}_\star\|_{\mathbf{V}_{t,p}}\|\mathbf{x}_{t,p}\|_{\mathbf{V}_{t,p}^{-1}} \\
&\overset{(i)}{\leq} \mathbb{1}[\mathcal{D}_t^c]\mathbb{E}\left[\mathbb{1}[\mathcal{E}']\left(\langle\tilde{\boldsymbol{\theta}}'_{t,p}, \mathbf{x}'_{t,p}\rangle - \bar{\boldsymbol{\theta}}_{t,p}^\top\bar{\mathbf{x}}_{t,p}\right)\Big|\mathcal{F}_{t,p-1}, \tilde{\boldsymbol{\theta}}'_{t,p} \in \boldsymbol{\Theta}_{t,p}, \mathcal{E}''_{t,p}\right] \\
&\quad + \mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\|\tilde{\boldsymbol{\theta}}_{t,p} - \boldsymbol{\theta}_\star\|_{\mathbf{V}_{t,p}}\|\mathbf{x}_{t,p}\|_{\mathbf{V}_{t,p}^{-1}} \\
&\overset{(ii)}{\leq} \mathbb{1}[\mathcal{D}_t^c]\mathbb{E}\left[\mathbb{1}[\mathcal{E}']\langle\tilde{\boldsymbol{\theta}}'_{t,p} - \bar{\boldsymbol{\theta}}_{t,p}, \mathbf{x}'_{t,p}\rangle\Big|\mathcal{F}_{t,p-1}, \tilde{\boldsymbol{\theta}}'_{t,p} \in \boldsymbol{\Theta}_{t,p}, \mathcal{E}''_{t,p}\right] + \\
&\quad \mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\|\tilde{\boldsymbol{\theta}}_{t,p} - \boldsymbol{\theta}_\star\|_{\mathbf{V}_{t,p}}\|\mathbf{x}_{t,p}\|_{\mathbf{V}_{t,p}^{-1}} \\
&\overset{(iii)}{\leq} \mathbb{1}[\mathcal{D}_t^c]\mathbb{E}\left[\mathbb{1}[\mathcal{E}']\|\tilde{\boldsymbol{\theta}}'_{t,p} - \bar{\boldsymbol{\theta}}_{t,p}\|_{\mathbf{V}_{t,p}}\|\mathbf{x}'_{t,p}\|_{\mathbf{V}_{t,p}^{-1}}\Big|\mathcal{F}_{t,p-1}, \tilde{\boldsymbol{\theta}}'_{t,p} \in \boldsymbol{\Theta}_{t,p}, \mathcal{E}''_{t,p}\right] \\
&\quad + \mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\|\tilde{\boldsymbol{\theta}}_{t,p} - \boldsymbol{\theta}_\star\|_{\mathbf{V}_{t,p}}\|\mathbf{x}_{t,p}\|_{\mathbf{V}_{t,p}^{-1}} \\
&\overset{(iv)}{\leq} 2\gamma_t(\delta)\mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\mathbb{E}\left[\|\mathbf{x}'_{t,p}\|_{\mathbf{V}_{t,p}^{-1}}\Big|\mathcal{F}_{t,p-1}, \tilde{\boldsymbol{\theta}}'_{t,p} \in \boldsymbol{\Theta}_{t,p}, \mathcal{E}''_{t,p}\right] + \\
&\quad \mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\|\tilde{\boldsymbol{\theta}}_{t,p} - \boldsymbol{\theta}_\star\|_{\mathbf{V}_{t,p}}\|\mathbf{x}_{t,p}\|_{\mathbf{V}_{t,p}^{-1}} \\
&\overset{(v)}{\leq} \frac{2\gamma_{t,p}(\delta)}{\frac{1}{4} - \frac{\delta}{2(t(P-1)+p)^2}}\mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\mathbb{E}\left[\|\mathbf{x}'_{t,p}\|_{\mathbf{V}_{t,p}^{-1}}\Big|\mathcal{F}_{t,p-1}\right] + \\
&\quad \gamma_{t,p}(\delta)\mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\|\mathbf{x}_{t,p}\|_{\mathbf{V}_{t,p}^{-1}}
\end{aligned}$$

Inequality (i) follows by Equation 4.23, (ii) by the definition of $\bar{x}_t$, (iii) is a consequence of Cauchy Schwartz, (iv) follows by the definition of $\mathcal{E}'$ and $\mathcal{E}''_{t,p}$, and (v) follows because $\|\mathbf{x}'_{t,p}\|_{\mathbf{V}^{-1}_{t,p}}$ is nonnegative and because by Lemma 6, it follows that $\mathbb{P}\left(\tilde{\boldsymbol{\theta}}'_{t,p} \in \boldsymbol{\Theta}_t, \mathcal{E}''_{t,p} | \mathcal{F}_{t,p-1}\right) \geq \frac{1}{4} - \frac{\delta}{2(t(P-1)+p)^2}$. The result follows.

□

Ancilliary Lemmas

In the proof of Theorem 3 we will also make use of the following supporting result:

Lemma 10. *Similar to Lemma 8, let $\tilde{\boldsymbol{\theta}}'_{t,p}$ copy of $\tilde{\boldsymbol{\theta}}_{t,p}$, equally distributed to $\tilde{\boldsymbol{\theta}}_{t,p}$ and independent conditionally on $\mathcal{F}_{t,p-1}$. We call $\mathbf{x}'_{t,p}$ to the resulting argmax action $\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}_{t,p}} \langle \mathbf{x}, \tilde{\boldsymbol{\theta}}'_{t,p} \rangle$. With probability at least $1-\delta$:*

$$\sum_{t=1}^{T}\sum_{p=1}^{P} \mathbb{E}[\|\mathbf{x}'_{t,p}\|_{\mathbf{V}^{-1}_{t,p}} | \mathcal{F}_{t,p-1}] \leq \sum_{t=1}^{T}\sum_{p=1}^{P} \|\mathbf{x}_{t,p}\|_{\mathbf{V}^{-1}_{t,p}} + \frac{2L}{\sqrt{\lambda}}\sqrt{TP\log(1/\delta)}$$

Proof. Define the martingale difference sequence $Z_{t,p} = \|\mathbf{x}_{t,p}\|_{\mathbf{V}^{-1}_{t,p}} - \mathbb{E}[\|\mathbf{x}'_{t,p}\|_{\mathbf{V}^{-1}_{t,p}} | \mathcal{F}_{t,p-1}]$ (the indexing is lexicographic over pairs (t,p)). It is easy to see that $|Z_{t,p}| \leq 2\frac{L}{\sqrt{\lambda}}$, since for all valid $\mathbf{x} \in \mathcal{X}_{t,p}$, $\|\mathbf{x}\|_{\mathbf{V}^{-1}_{t,p}} \leq \frac{L}{\sqrt{\lambda}}$ for all t,p pairs. Consequently a simple use of Hoeffding bound yields the result.

□

Proof of Theorem 3

We proceed to prove the general version of Theorem 3:

Theorem 6. *Let Assumptions 1, 2 and 3 hold and $\mathcal{D}_t$ denote the event that round t is a doubling round (see Condition 1). Then the regret of both Algorithms 3 and 4 satisfy ,*

$$\mathcal{R}(T,P) \leq \tilde{O}\Bigg(LSP \cdot \sum_{t=1}^{T} \mathbb{1}[\mathcal{D}_t] + \epsilon TP + 4\gamma_T(\delta)\left(\sqrt{dTP\left(1+\frac{L^2}{\lambda}\right)\ln\left(\frac{d\lambda+TPL}{d\lambda}\right)} + \frac{2L}{\sqrt{\lambda}}\sqrt{TP\log(1/\delta)}\right)\Bigg)$$

with probability at least $1-3\delta$, whenever $\delta \leq \frac{1}{6}$. Here

$$\gamma_T(\delta) = \sqrt{2}(\sqrt{\beta_T(\delta)} + \sqrt{(T-1)P\epsilon})\left(\sqrt{d} + 2\sqrt{\log\left(\frac{T(P-1)+P}{\delta}\right)} + 1\right).$$

Proof of Theorem 3. The regret in terms of $f(\cdot)$ can be linearized at the cost of an additive $2\epsilon\sqrt{d}TP$ as in the Proof of Theorem 2. After this we can decompose the regret for Algorithm 3, splitting on the event each round over the event $\mathcal{D}_t$,

$$\begin{aligned}\mathcal{R}(T,P) &= \sum_{t=1}^{T}\left(\sum_{p=1}^{P}\underbrace{\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star\rangle}_{r_{t,p}}\right) \\ &= \sum_{t=1}^{T}\mathbb{1}[\mathcal{D}_t]\left(\sum_{p=1}^{P}\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star\rangle\right) + \sum_{t=1}^{T}\mathbb{1}[\mathcal{D}_t^c]\left(\sum_{p=1}^{P}\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star\rangle\right)\end{aligned}$$

The first term can be bounded using Assumptions 2 and 3 along with the Cauchy-Schwarz inequality which gives $\left(\sum_{p=1}^{P}\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star\rangle\right) \leq 2LSP$ so:

$$\sum_{t=1}^{T}\mathbb{1}[\mathcal{D}_t]\left(\sum_{p=1}^{P}\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star\rangle\right) \leq 2LSP\sum_{t=1}^{T}\mathbb{1}[\mathcal{D}_t].$$

This holds for any sequence of $\mathbf{x}_{t,p}$ chosen by the doubling round routine DR. We turn our attention to the second term.

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{1}[\mathcal{D}_t^c]\left(\sum_{p=1}^{P}\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star\rangle\right) \\ &= \sum_{t=1}^{T}\left(\sum_{p=1}^{P}\mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star\rangle\right) + \sum_{t=1}^{T}\left(\sum_{p=1}^{P}\mathbb{1}[\mathcal{D}_t^c \cap (\mathcal{E}')^c]\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star\rangle\right)\end{aligned}$$

Notice that:

$$\sum_{t=1}^{T}\sum_{p=1}^{P}\mathbb{1}[\mathcal{D}_t^c \cap (\mathcal{E}')^c]\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star\rangle \leq \sum_{t=1}^{T}\sum_{p=1}^{P}2\mathbb{1}[\mathcal{D}_t^c \cap (\mathcal{E}')^c]LS \leq 2LSTP\mathbb{1}[(\mathcal{E}')^c]$$

And therefore we can forget this term when we condition on $\mathcal{E}'$, an event that occurs with probability at least $1-2\delta$ (recall $\mathcal{E}'$ as the event from Lemma 6).

It remains to bound the first term of Equation 4.24.

$$\begin{aligned}
\sum_{t=1}^{T}\sum_{p=1}^{P} \mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\langle \mathbf{x}^\star_{t,p} - \mathbf{x}_{t,p}, \boldsymbol{\theta}^\star\rangle &\overset{(i)}{\leq} \sum_{t=1}^{T}\sum_{p=1}^{P} \frac{8}{1-3\delta}\gamma_{t,p}(\delta)\mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\mathbb{E}\left[\|\mathbf{x}'_{t,p}\|_{\mathbf{V}_{t,p}^{-1}}|\mathcal{F}_{t,p-1}\right] + \\
&\quad \gamma_{t,p}(\delta)\mathbb{1}[\mathcal{D}_t^c \cap \mathcal{E}']\|\mathbf{x}_{t,p}\|_{\mathbf{V}_{t,p}^{-1}} \\
&\overset{(ii)}{\leq} \sum_{t=1}^{T}\sum_{p=1}^{P} \frac{8}{1-3\delta}\gamma_{t,p}(\delta)\mathbb{E}\left[\|\mathbf{x}'_{t,p}\|_{\mathbf{V}_{t-1,p-1}^{-1}}|\mathcal{F}_{t-1,p-1}\right] \\
&\quad + \gamma_{t,p}(\delta)\|\mathbf{x}_{t,p}\|_{\mathbf{V}_{t,p}^{-1}} \\
&\overset{(iii)}{\leq} \left(\frac{8}{1-3\delta}+1\right)\gamma_{T,P}(\delta)\left(\sum_{t=1}^{T}\sum_{p=1}^{P}\|\mathbf{x}_{t,p}\|_{\mathbf{V}_{t-1,p-1}^{-1}}\right. \\
&\quad \left. + \frac{2L}{\sqrt{\lambda}}\sqrt{TP\log(1/\delta)}\right) \\
&\leq 16\gamma_T(\delta)\left(\sqrt{TP}\sum_{t=1}^{T}\sum_{p=1}^{P}\|\mathbf{x}_{t,p}\|^2_{\mathbf{V}_{t-1,p-1}^{-1}}\right. \\
&\quad \left. + \frac{2L}{\sqrt{\lambda}}\sqrt{TP\log(1/\delta)}\right) \\
&\leq 16\gamma_T(\delta)\left(\sqrt{dTP\left(1+\frac{L^2}{\lambda}\right)\ln\left(\frac{d\lambda+TPL}{d\lambda}\right)}\right. \\
&\quad \left. + \frac{2L}{\sqrt{\lambda}}\sqrt{TP\log(1/\delta)}\right)
\end{aligned}$$

Inequality (i) holds by Lemma 8 and the assumption that $\delta \leq \frac{1}{6}$. Inequality (ii) holds because all terms $\|\mathbf{x}'_{t,p}\|_{\mathbf{V}_{t,p}^{-1}}$ and $\|\mathbf{x}_{t,p}\|_{\mathbf{V}_{t,p}^{-1}}$ are nonnegative. Inequality (iii) holds with probability at least $1-\delta$ and is a consequence of lexicographic monotonicity (in t,p) of $\gamma_{t,p}(\delta)$ and Lemma 10. The last two inequalities are a simple consequence of the determinant lemma (Lemma 11). □

Auxiliary Results

Here we summarize the self-normalized vector martingale inequality used to establish the confidence ball for the least-squares estimator in a well-specified linear model,

$$r_{t,p} = \mathbf{x}_{t,p}^\top \boldsymbol{\theta}^\star + \xi_{t,p}. \tag{4.24}$$

Here $\xi_{t,p}$ is an i.i.d. noise process.

Theorem 7. *[Theorem 1 in [1]] For all $t \in \mathbb{N}$:*

$$\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,1}} \leq \sqrt{\beta_t(\delta)}$$

with probability at least $1-\delta$. Moreover, on this event by definition,

$$\boldsymbol{\theta}^\star \in \mathcal{C}(\hat{\boldsymbol{\theta}}_t, \mathbf{V}_{t,1}, \beta_t(\delta)).$$

We can now prove a generalization of this result which applies to the analysis of the lazy LinUCB algorithm in a well-specified model.

Theorem 8. *Let $N \subseteq [T]$ be the set of rounds which are not doubling rounds (see Condition 1). Then,*

$$\forall t \in N, \quad \|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,p}} \leq \sqrt{2}\sqrt{\beta_t(\delta)}$$

with probability at least $1-\delta$. Moreover, on this event by definition,

$$\boldsymbol{\theta}^\star \in \mathcal{C}(\hat{\boldsymbol{\theta}}_t, \mathbf{V}_{t,p}, 2\beta_t(\delta)).$$

Proof. Let $N \subseteq \mathbb{N}$ be the set of rounds which are not doubling rounds. Then if $t \in N$,

$$\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,p}} \leq \sqrt{2}\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,1}}$$

from the definition in Condition 1. Hence,

$$\mathbb{P}[\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,p}} \geq \sqrt{2}\sqrt{\beta_t(\delta)}, t \in N] \leq \mathbb{P}[\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,1}} \geq \sqrt{\beta_t(\delta)}, t \in N] \leq$$
$$\mathbb{P}[\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,1}} \geq \sqrt{\beta_t(\delta)}, \forall t \in \mathbb{N}] \leq \delta.$$

where the final inequality follows by Theorem 7.

□

Define $\mathcal{E}$ to the $1-\delta$ probability event defined in Theorem 8:

$$\mathcal{E} := \{\forall t \in N, \quad \|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,p}} \leq \sqrt{2}\sqrt{\beta_t(\delta)}\}$$

We are now in a position to prove generalizations of these results which provide valid confidence sets for the linear regression estimator in *misspecified* models. In summary, the confidence sets are modified with a growing, additive correction to accommodate the bias arising from the misspecification.

Theorem 9. *If the rewards are generated from a model satisfying Assumption 3, then for all $t \in \mathbb{N}$:*

$$\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,1}} \leq \sqrt{\beta_t(\delta)} + \sqrt{(t-1)P}\epsilon$$

with probability at least $1-\delta$. Moreover, on this event by definition,

$$\boldsymbol{\theta}^\star \in \mathcal{C}(\hat{\boldsymbol{\theta}}_t, \mathbf{V}_{t,1}, \beta_t(\delta), \epsilon).$$

Proof. The argument uses a bias-variance decomposition. First, define the linearized reward $\tilde{r}_{a,b} = \mathbf{x}_{a,b}^\top \boldsymbol{\theta}^\star + \xi_{a,b}$ and linear estimator using these rewards as $\tilde{\boldsymbol{\theta}} = \mathbf{V}_{t,1}^{-1}(\sum_{a=1}^{t-1}\sum_{b=1}^{P}\mathbf{x}_{a,b}\tilde{r}_{a,b})$. By definition, $r_{a,b} - \tilde{r}_{a,b} = \epsilon_{a,b}$ (which all satisfy $|\epsilon_{a,b}| \leq \epsilon$ uniformly for all $\mathbf{x}_{a,b}$ by assumption). Then,

$$\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star = \tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star = \tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star + \mathbf{V}_{t,0}^{-1}(\sum_{a=1}^{t-1}\sum_{b=1}^{P}\mathbf{x}_{a,b}\epsilon_{a,b}) \implies$$

$$\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,1}} \leq \|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,1}} + \|\sum_{a=1}^{t-1}\sum_{p=1}^{P}\mathbf{x}_{a,b}\epsilon_{a,b}\|_{\mathbf{V}_{t,1}^{-1}}$$

The first term can be bounded by $\sqrt{\beta_t(\delta)}$ with probability $1-\delta$ exactly by using Theorem 7. Using the projection bound in [99, Lemma 8] it follows that,

$$\|\sum_{a=1}^{t-1}\sum_{p=1}^{P}\mathbf{x}_{a,b}\epsilon_{a,b}\|_{\mathbf{V}_{t,1}^{-1}} \leq \sqrt{(t-1)P}\epsilon \tag{4.25}$$

since $|\epsilon_{a,b}| \leq \epsilon$ uniformly for all a, b.

□

The analogue for the lazy confidence set follows similarly,

Theorem 10. *Let $N \subseteq [T]$ be the set of rounds which are not doubling rounds (see Condition 1). Then, If the rewards are generated from a model satisfying Assumption 3, for all $t \in N$:*

$$\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,p}} \leq \sqrt{2}(\sqrt{\beta_t(\delta)} + \sqrt{(t-1)P}\epsilon)$$

with probability at least $1-\delta$. Moreover, on this event by definition,

$$\boldsymbol{\theta}^\star \in \mathcal{C}(\hat{\boldsymbol{\theta}}_t, \mathbf{V}_{t,1}, 2\beta_t(\delta), 2\epsilon).$$

Proof. First, if $t \in N$,

$$\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,p}} \leq \sqrt{2}\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,1}}$$

from the definition in Condition 1. The remainder of the argument follows as in the previous result,

$$\mathbb{P}[\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,p}} \geq \sqrt{2}(\sqrt{\beta_t(\delta)} + \sqrt{(t-1)P}\epsilon), t \in N] \leq \mathbb{P}[\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,1}} \geq$$
$$\sqrt{\beta_t(\delta)} + \sqrt{(t-1)P}\epsilon, t \in N] \leq \mathbb{P}[\|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|_{\mathbf{V}_{t,1}} \geq \sqrt{\beta_t(\delta)}, \forall t \in \mathbb{N}] \leq \delta.$$

where the final inequality follows by Theorem 7, since $\tilde{\boldsymbol{\theta}} = \mathbf{V}_{t,1}^{-1}(\sum_{a=1}^{t-1}\sum_{b=1}^{P}\mathbf{x}_{a,b}\tilde{r}_{a,b})$ is the estimator utilizing the linearized rewards. □

Next we recall the elliptical potential lemma which control the volumetric growth of the space spanned by sequence of covariance matrices.

Lemma 11. *[Lemma 19.4 in [57]] Let $\mathbf{V}_0 \in \mathbb{R}^{d\times d}$ be a positive-definite matrix, $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ be a sequence of vectors with $\|\mathbf{x}_i\| \le L$ for all $i \in [n]$, and $\mathbf{V}_n = \mathbf{V}_0 + \sum_{s\le n} \mathbf{x}_s\mathbf{x}_s^\top$. Then,*

$$\sum_{s=1}^{n} \log(1 + \|\mathbf{x}_s\|^2_{\mathbf{V}_{s-1}^{-1}}) = \log\left(\frac{\det \mathbf{V}_n}{\det \mathbf{V}_0}\right)$$

$$\log\left(\frac{\det \mathbf{V}_n}{\det \mathbf{V}_0}\right) \le d\log\left(\frac{\mathrm{tr}(\mathbf{V}_0) + nL^2}{d}\right) - \log(\det \mathbf{V}_0)$$

Finally, we state a prove a simple fact from linear algebra.

Lemma 12. *If $\mathbf{A} \succeq \mathbf{B} \succ 0$,*

$$\forall \mathbf{x} \neq 0 \in \mathbb{R}^d, \quad \frac{\mathbf{x}^\top \mathbf{A}\mathbf{x}}{\mathbf{x}^\top \mathbf{B}\mathbf{x}} \le \frac{\det(\mathbf{A})}{\det(\mathbf{B})}.$$

Proof of Lemma 12. To this end, let $\mathbf{x} = \mathbf{B}^{-1/2}\mathbf{y}$ for $\mathbf{y} \in \mathbb{R}^d$. Then note that,

$$\sup_{\mathbf{x}\neq 0} \frac{\mathbf{x}^\top \mathbf{A}\mathbf{x}}{\mathbf{x}^\top \mathbf{B}\mathbf{x}} = \|\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}\|_2$$

by the definition of the operator norm. Similarly, we can rewrite $\frac{\det(\mathbf{A})}{\det(\mathbf{B})} = \det(\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2})$. The claim then follows because all the eigenvalues of $\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}$ are ≥ 1. Note that

$$\inf_{\mathbf{x}:\|\mathbf{x}\|_2=1} \mathbf{x}^\top \mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}\mathbf{x} = 1 + \mathbf{x}^\top \mathbf{B}^{-1/2}(\mathbf{A}-\mathbf{B})\mathbf{B}^{-1/2}\mathbf{x} \ge 1$$

since $\mathbf{A} \succeq \mathbf{B}$. □

Proofs in Section 4.3

We now present the proof of Corollary 5.

Proof of Corollary 5. As a consequence of Theorem 2 we have for any choice of doubling round routine that the regret of Algorithm 1 and Algorithm 2 obeys,

$$\mathbb{1}[\mathcal{E}]\mathcal{R}(T,P) \le \mathbb{1}[\mathcal{E}]8\cdot\Bigg(\underbrace{LSP\sum_{t=1}^{T}\mathbb{1}[\mathcal{D}_t]}_{A} + \underbrace{\sqrt{TP}\max(\sqrt{\beta_T(\delta)} + \sqrt{TP}\epsilon, LS)\cdot\sqrt{d\log\left(1+\frac{TPL^2}{\lambda}\right)} + \epsilon TP}_{B}\Bigg)$$

where $\sqrt{\beta_T(\delta)} \leq R\sqrt{d\log\left(\frac{1+TPL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}S$. This inequality holds on the event $\mathcal{E}$ which occurs with probability at least $1-\delta$ for both Algorithm 1 and Algorithm 2. Now we introduce an additional event $\mathcal{G} = \{\sum_{t=1}^{T} \mathbb{1}[\mathcal{D}_t] \geq \lceil \frac{10^5}{P}\frac{L^2\pi_{\max}^2}{\pi_{\min}^4}\log(\frac{4d}{\delta})\rceil\}$. Then we can consider two cases,

- First,

$$\mathbb{1}[\mathcal{G}^c](A+B) \leq \mathbb{1}[\mathcal{G}^c]\left(LS\left\lceil\frac{10^5}{P}\frac{L^2\pi_{\max}^2}{\pi_{\min}^4}\log(\frac{4d}{\delta})\right\rceil + B\right)$$

  simply by definition of the event.

- Second, by definition of the $\mathbf{V}_{t,p}$ and $\mathcal{G}$ all rounds which are not doubling rounds (denoted by the set N) lead to randomly generated covariates being used to estimate the covariance. Let $r_1 \in N$ be the first doubling round to exceed the threshold $\lceil\frac{10^5}{P}\frac{L^2\pi_{\max}^2}{\pi_{\min}^4}\log(\frac{4d}{\delta})\rceil$. We claim (with high probability) there can be no more doubling rounds after round r_1. This follows since first by Lemma 13 on the event $\mathcal{G}$,

$$\|\sum_{a\in N}\sum_{p=1}^{P}\mathbf{x}_{a,p}\mathbf{x}_{a,p}^\top - (\mathbf{\Sigma}_{\pi_{a,p}} + \mu_{a,p}\mu_{a,p}^\top))\|_2 \leq \begin{cases} \frac{1}{10}\pi_{\min}^2|N|P & \text{when } |N|>1 \\ \frac{1}{10}\pi_{\min}^2 P & \text{when } |N|=1 \end{cases}$$

  with probability at least $1-\delta$ (denote this further event $\mathcal{C}$). This previous inequality follows by considering the two separate cases where $|N|>1$ and $|N|=1$ respectively. Thus in any round $t > r_1$ we must have with probability $1-\delta$,

$$\mathbf{V}_{t,1} \succeq \sum_{a\in N}\sum_{p=1}^{P}(\mathbf{\Sigma}_{\pi_{a,p}} + \mu_{i,p}\mu_{i,p}^\top) - \frac{\pi_{\min}^2|N|P}{10}\mathbf{I} \succeq \frac{9}{10}|N|P\pi_{\min}^2\mathbf{I} \quad \text{when } |N|>1$$

$$\mathbf{V}_{t,1} \succeq P(\mathbf{\Sigma}_{\pi_{1,p}} + \mu_{1,p}\mu_{1,p}^\top) - \frac{\ell^2 P}{10}\mathbf{I} \succeq \frac{9}{10}P\ell^2\mathbf{I} \quad \text{when } |N|=1$$

  using Definition 2. In summary for all $|N| \geq 1$ we then have that

$$\mathbf{V}_{t,1} \succeq \frac{9}{10}|N|P\pi_{\min}^2\mathbf{I}.$$

  If, additionally it is the case that $|N| \geq \lceil\frac{10}{9}\frac{\chi^2}{\pi_{\min}^2}\rceil$, let r_2 be the first round after which this occurs. Then it follows that for all $t > \max(r_1, r_2)$,

$$\mathbb{1}[\mathcal{C}](\sum_{p=1}^{P}\mathbf{x}_{t,p}\mathbf{x}_{t,p}^\top) \preceq \mathbb{1}[\mathcal{C}]P\chi^2\mathbf{I} \preceq \frac{9}{10}|N|P\pi_{\min}^2\mathbf{I} \preceq \mathbf{V}_{t,1}, \tag{4.26}$$

  so no $t > \max(r_1, r_2)$ can be a doubling round on this event. Concluding we have that,

$$\mathbb{1}[\mathcal{G}]\mathbb{1}[\mathcal{C}]A \leq LSP \cdot \max(\lceil\frac{10^5}{P}\frac{L^2\pi_{\max}^2}{\pi_{\min}^4}\log(\frac{4d}{\delta})\rceil, \lceil\frac{10}{9}\frac{\chi^2}{\pi_{\min}^2}\rceil).$$

Together, we obtain,

$$\mathbb{1}[\mathcal{G}]\mathbb{1}[\mathcal{C}](A+B) \leq \mathbb{1}[\mathcal{G}]\mathbb{1}[\mathcal{C}]\left(LSP \cdot \max(\lceil \frac{10^5}{P}\frac{L^2\pi_{\max}^2}{\pi_{\min}^4}\log(\frac{4d}{\delta})\rceil, \lceil \frac{10}{9}\frac{L^2}{\ell^2}\rceil) + B\right)$$

Assembling and summing these two cases, then shows that,

$$\mathbb{1}[\mathcal{C}](A+B) \leq (LSP \cdot \max(\lceil \frac{10^5}{P}\frac{L^2\pi_{\max}^2}{\pi_{\min}^4}\log(\frac{4d}{\delta})\rceil, \lceil \frac{10}{9}\frac{\chi^2}{\ell^2}\rceil) + B).$$

So it follows that,

$$\mathbb{1}[\mathcal{E}]\mathbb{1}[\mathcal{C}]\mathcal{R}(T,P) \leq \mathbb{1}[\mathcal{E}]\mathbb{1}[\mathcal{C}] \cdot 8 \cdot (LSP \cdot \max(\lceil \frac{10^5}{P}\frac{L^2\pi_{\max}^2}{\pi_{\min}^4}\log(\frac{4d}{\delta})\rceil, \lceil \frac{10}{9}\frac{\chi^2}{\ell^2}\rceil) +$$
$$\sqrt{TP}\max(\sqrt{\beta_T(\delta)} + \sqrt{TP}\epsilon, LS) \cdot \sqrt{d\log\left(1 + \frac{TP\chi^2}{\lambda}\right)} + \epsilon TP)$$

where both $\mathcal{E}$ and $\mathcal{C}$ hold with probability $1-\delta$. We can simplify the first term to,

$$LSP \cdot \max(\lceil \frac{10^5}{P}\frac{L^2\pi_{\max}^2}{\pi_{\min}^4}\log(\frac{4d}{\delta})\rceil, \lceil \frac{10}{9}\frac{\chi^2}{\ell^2}\rceil)$$
$$\leq \tilde{O}(LSP \cdot (\frac{10^5}{P}\frac{L^2\pi_{\max}^2}{\pi_{\min}^4}\log(\frac{4d}{\delta}) + 1 + \frac{\chi^2}{\ell^2})) \leq \tilde{O}(LS(\frac{L^4}{\ell^4} + P\frac{\chi^2}{\ell^2}))$$

Hiding logarithmic factors, this implies that,

$$\mathcal{R}(T,P) \leq \tilde{O}\left(R\left((\frac{L^2\|\mathbf{\Sigma}_\pi\|}{\ell^4} + P\frac{\chi^2}{\ell^2})\sqrt{\text{SNR}} + d\sqrt{TP} + \frac{\epsilon}{R}TP\right)\right)$$

with probability at least $1-2\delta$.

An identical argument establishes the result for the Thompson sampling algorithms save with Thompson sampling regret $\mathcal{R}(T,P)$ used in place of the LinUCB regrets in the previous argument. □

We now present the matrix concentration result we use,

Lemma 13. *Let Definition 2 and Assumption 2 hold and consider N i.i.d. copies of sets (with P elements) sampled from Algorithm 5, labeled as $\{\mathbf{x}_{i,p}\}_{i=1,p=1}^{N,P}$. Then,*

$$\|\frac{1}{NP}(\sum_{i=1}^{N}\sum_{p=1}^{P}\mathbf{x}_{i,p}\mathbf{x}_{i,p}^\top - (\mathbf{\Sigma}_{\pi_{i,p}} + \mu_{i,p}\mu_{i,p}^\top))\|_2 \leq 12\left(L\sqrt{\pi_{\max}^2}\sqrt{\frac{\log(4d/\delta)}{NP}} + \frac{L^2\log(4d/\delta)}{NP}\right)$$

with probability at least $1-\delta$.

Proof. We first center the expression around its mean $\mathbb{E}_{\pi_{i,p}}[\mathbf{x}_{i,p}] = \mu_{i,p}$. That is,

$$\|\frac{1}{NP}\sum_{i=1}^{N}\sum_{p=1}^{P}\mathbf{x}_{i,p}\mathbf{x}_{i,p}^{\top} - (\mathbf{\Sigma}_{\pi_{i,p}} + \mu_{i,p}\mu_{i,p}^{\top})\|_2 \leq$$
$$\|\frac{1}{NP}\sum_{i=1}^{N}\sum_{p=1}^{P}(\mathbf{x}_{i,p}-\mu_{i,p})(\mathbf{x}_{i,p}-\mu_{i,p})^{\top} - \mathbf{\Sigma}_{\pi_{i,p}}\|_2 + 2\|\frac{1}{NP}\sum_{i=1}^{N}\sum_{p=1}^{P}\mu_{i,p}(\mathbf{x}_{i,p}-\mu_{i,p})^{\top} - \mu_{i,p}\mu_{i,p}^{\top}\|_2$$

We now apply the matrix Bernstein inequality to control this first term [89, Theorem 1.6.2]. Note that for all i, $\|(\mathbf{x}_{i,p}-\mu_{i,p})(\mathbf{x}_{i,p}-\mu_{i,p})^{\top}\| \leq 2L^2$ and the matrix variance is bounded by

$$\|\mathbb{E}[(\mathbf{x}_{i,p}-\mu_{i,p})(\mathbf{x}_{i,p}-\mu_{i,p})^{\top} - \mathbf{\Sigma}_{\pi_{i,p}})^2]\|_2 \leq \cdot\|\mathbb{E}[\|\mathbf{x}_{i,p}-\mu\|_2^2(\mathbf{x}_{i,p}-\mu)(\mathbf{x}_{i,p}-\mu)^{\top})]\|_2 \leq 2L^2\pi_{\max}^2$$

. Thus we obtain,

$$\|\frac{1}{NP}\sum_{i=1}^{N}\sum_{p=1}^{P}(\mathbf{x}_{i,p}-\mu_{i,p})(\mathbf{x}_{i,p}-\mu_{i,p})^{\top} - \mathbf{\Sigma}_{\pi_{i,p}}\|_2 \leq 4\left(L\sqrt{\pi_{\max}^2}\sqrt{\frac{\log(4d/\delta)}{NP}} + \frac{L^2\log(4d/\delta)}{NP}\right) \tag{4.27}$$

with probability at least $1-\delta/2$. For the second term note by Jensen's inequality that $\|\mu_{i,p}\|_2 = \|\mathbb{E}[\mathbf{x}_{i,p}]\|_2 \leq \mathbb{E}[\|\mathbf{x}_{i,p}\|_2] \leq L$ since $\|\mathbf{x}_{i,p}\|_2 \leq L$. An identical calculation before to bound almost surely bound this term and its matrix variance we have that,

$$\|\frac{1}{NP}\sum_{i=1}^{N}\sum_{p=1}^{P}\mu_{i,p}(\mathbf{x}_{i,p}-\mu_{i,p})^{\top} - \mu_{i,p}\mu_{i,p}^{\top}\|_2 \leq 4\left(L\sqrt{\pi_{\max}^2}\sqrt{\frac{\log(4d/\delta)}{NP}} + \frac{L^2\log(4d/\delta)}{NP}\right)$$

by the matrix Bernstein inequality with probability at least $1-\delta/2$. Summing the terms and applying a union bound over the events on which they hold gives the result. □

Proofs in Section 4.4

Here we include the proof of the main lower bound.

Proof of Theorem 4. The proof follows by first noting that in each of the instances claimed a single, fixed global context vector is used for all time and processors. Hence the parallel to sequential regret reduction established in Proposition 12 is applicable. Thus it suffices to establish the lower bounds for $R(TP,1)$ in lieu of $R(T,P)$ for the instances claimed. The first term/result is an immediate consequence of Lemma 14. The second term/result we can obtain from Lemma 16. For the validity of the results, we inherit the constraints $d \geq \lceil 8\log(m)L^2/\epsilon^2\rceil$ and $S \geq \|\boldsymbol{\theta}^{\star}\|_2 = \frac{\epsilon}{L}\sqrt{\frac{d-1}{8\log(m)}} \implies \epsilon^2 \leq \frac{8(LS)^2\log(m)}{d-1}$. If we take $d = \lceil 8\log(m)/\epsilon^2\rceil$ then the second constraint reduces too $\epsilon^2 \leq \frac{8(LS)^2\log(m)}{8\log(m)/\epsilon^2} \implies LS \geq 1$.

□

Parallel to Sequential Reduction

Here we establish the reduction from parallel regret to sequential regret when considering parallel linear bandits where there is a single fixed action set/context set across all processors at a given time. So $\mathcal{X}_{t,p} = \mathcal{X}_t$ across all processors $p \in [P]$. Additionally, we assume as in the preamble that the reward of an action $\mathbf{x}$ is determined by $r = f(\mathbf{x}) + \epsilon$–so the law of the rewards is completely specified by a mean reward function f and mean-zero noise distribution ϵ.

We formalize the reduction from parallel to sequential bandits by first defining the canonical bandit environment. We consider the bandit instance to be indexed by the law of the rewards and the sequence of context sets.

Sequential In a model of purely sequential interaction we consider instances defined by two ingredients:

- the conditional distribution of the policy $\pi_t(\cdot|\mathbf{x}_{i<t}, \mathcal{X}_{i\leq t}, r_{i<t})$.
- the sequence of reward distributions $\nu_s \equiv \mathbb{P}_t(\cdot|x_{i<t}, \mathcal{X}_{i\leq t}, r_{i<t}, f) \equiv \mathbb{P}_{\mathbf{x}_t}(\cdot|f)$ for selected actions and the sequence of presented contexts $\mathcal{X}_t$.

Parallel In a model of parallel interaction, there are two ingredients:

- the conditional distribution of policy $\psi_{t,p}(\cdot|x_{i,j<t,p}, \mathcal{X}_{i\leq t}, r_{i,j<t,p})$.
- the sequence of reward distributions $\nu_p \equiv \mathbb{P}_{t,p}(\cdot|\mathbf{x}_{i,j<t,p}, \mathcal{X}_{i\leq t}, r_{i,j<t,1}) \equiv \mathbb{P}_{\mathbf{x}_{t,p}}(\cdot|f)$ for selected actions and sequence of selected contexts $\mathcal{X}_t$ (which for fixed t are equal across all $p \in [P]$).

To formalize the reduction we make the following claim:

Proposition 11. *If we consider the lexicographic ordering for $t, p \in [T, P]$, then for any sequence of parallel policy-environment interactions with law $(\psi_{t,p}(\cdot), \mathbb{P}_{t,p}(\cdot|f))$ and presented context sets $\mathcal{X}_t$ (identical across $p \in [P]$), there exists a corresponding coupling to a purely sequential bandit environment $(\pi_m(\cdot), \mathbb{P}_m(\cdot|f))$ for $m \in [TP]$ and sequence of context sets $\mathcal{X}_m = \mathcal{X}_t$ for $m \in [tP, (t+1)P]$ with an identical distribution.*

Proof. We can construct the sequential environment inductively from the sequence of parallel interaction by coupling. To see this consider the first round of parallel interactions which are described by the measure,

$$\Pi_{p=1}^{P} \psi_{1,p}(\cdot|\mathbf{x}_{i,j<1,p}, \mathcal{X}_1, r_{i,j<1,1}) \mathbb{P}_{\mathbf{x}_{1,p}}(\cdot|f).$$

By defining the measure over sequential policy-environment interactions,

$$\Pi_{m=1}^{P} \pi_m(\cdot|\mathbf{x}_{a<m} \mathcal{X}_m, r_{a<m}) \mathbb{P}_{\mathbf{x}_m}(\cdot|f)$$

we can set, $\pi_m(\cdot|\mathbf{x}_{a<m}\mathcal{X}_{a<m}, r_{a<m}) = \psi_{1,p}(\cdot|\mathbf{x}_{i,j<1,p}\mathcal{X}_1, r_{i,j<1,1})$ by ignoring the conditioning on the further contexts and reward information in the sequential interaction to enforce equality of the policies. Since the policies are tamen to be identical, by coupling the randomness between the sequential and parallel policies/environments, the sequence of selected actions will be identical (this uses the fact the context sets in the parallel blocm and sequential interaction can be tamen equal). Inductively proceeding with the construction over the blocms of parallel interaction completes the argument. □

With this claim in hand the reduction follows since the expected regret of an algorithm over an environment (in expectation) is determined by the law of the policy-environment interactions. Before beginning we first introduce the following notation for the expected parallel regret (further indexed by policy and environment,

$$R_{\psi,\nu_p}(T,P) = \sum_{t=1}^{T}\sum_{p=1}^{P} \max_{\mathbf{x}\in\mathcal{X}_t} \mu(\mathbf{x}) - \mathbb{E}_{\psi,\nu_p}[\mu(\mathbf{x})] \tag{4.28}$$

where ψ denotes the parallel policy and ν_p the parallel environment indexed by the mean reward and sequence of context sets. Similarly, the sequential regret can be defined analogously as,

$$R_{\pi,\nu_s}(TP,1) = \sum_{a=1}^{TP} \max_{\mathbf{x}\in\mathcal{X}_a} \mu(\mathbf{x}) - \mathbb{E}_{\pi,\nu_s}[\mu(\mathbf{x})] \tag{4.29}$$

where π denotes the sequential policy and ν_s the sequential environment indexed by the mean reward and sequence of context sets which *are fixed to be the same over consecutive blocms of length P.*

Proposition 12. *Consider both a parallel bandit policy/environment class and sequential bandit policy/environment class as defined in Proposition 11. Then,*

$$\inf_{\psi}\sup_{\nu_p} R_{\psi,\nu_p}(T,P) \geq \inf_{\pi}\sup_{\nu_s} R_{\pi,\nu_s}(TP,1) \tag{4.30}$$

where the infima over ψ is tamen place over the class of all parallel policies and the infima on the right hand side is tamen over the class of all sequential policies.

Proof. By the the preceding claim in Proposition 11 every pair of parallel (ψ,ν_p) measures can be reproduced by a corresponding sequential measure (π,ν_s). Let the set of such induced sequential policies be $\mathcal{P}$. So the pointwise inequality,

$$R_{\psi,\nu_p}(T,P) = R_{\pi,\nu_s}(TP,1) \tag{4.31}$$

holds by simply re-indexing the summation in lexigraphic order since the expectations are identical. Hence, for a fixed ψ (and induced π) the equality also holds after taming a supremum over the same indexing set on both sides (parameterized by f and $\mathcal{X}_t$),

$$\sup_{\nu_p} R_{\psi,\nu_p}(T,P) = \sup_{\nu_s} R_{\pi,\nu_s}(TP,1) \tag{4.32}$$

Now taming an infima over the policy class ψ and equivalent induced sequential policy class shows,

$$\inf_{\psi} \sup_{\nu_p} R_{\psi,\nu_p}(T,P) = \inf_{\pi \in \mathcal{P}} \sup_{\nu_s} R_{\pi,\nu_s}(TP,1) \geq \inf_{\pi} \sup_{\nu_1} R_{\pi,\nu_s}(TP,1) \tag{4.33}$$

by relaxing the final infima to tame place over the class of all sequential policies. □

Unit Ball Lower Bound

Here we record the lower bound over a fixed action set for the unit ball for a sequential bandit instance. The proof is an immediate generalization of [57, Theorem 24.2] with the scales restored. Throughout this section we assume that the rewards are generated as,

$$r_{t,p} = \mathbf{x}_{t,p}^{\top} \boldsymbol{\theta}^{\star} + \xi_{t,p} \tag{4.34}$$

where $\xi_{t,p} \sim \mathcal{N}(0, R^2)$.

Lemma 14. *Let the fixed action set $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 \leq L\}$ and parameter set $\Theta = \{\pm\Delta\}^d$ for $\Delta = \frac{R\sqrt{d}}{3\sqrt{2T}L}$. If $T \geq d \max(1, \frac{1}{3\sqrt{2}\sqrt{\text{SNR}}})$, then for any policy π, there is a vector $\boldsymbol{\theta}^{\star} \in \Theta$ such that:*

$$R_{\pi,(\mathcal{X},\boldsymbol{\theta})}(T,1) \geq \Omega\left(Rd\sqrt{T}\right).$$

Proof of Lemma 14. Let $\mathcal{A} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 \leq L\}$ and $\boldsymbol{\theta} \in \mathbb{R}^d$ such that $\|\boldsymbol{\theta}\|_2^2 = S^2$. Let $\Delta = \frac{R\sqrt{d}}{3\sqrt{2T}L}$ and $\boldsymbol{\theta} \in \{\pm\Delta\}^d$ and for all $i \in [d]$ define $\tau_i = \min(T, \min(t : \sum_{s=1}^{t} \mathbf{x}_{s,i}^2 \geq \frac{\alpha n}{d}\}$. We will set $\alpha = L^2$ at the end of the proof but in order to mame the derivations clearer and easier to read we will meep this α explicit. Then for any policy π:

$$\begin{aligned}
R_{\pi,(\mathcal{X},\boldsymbol{\theta})}(T,1) &= \Delta \mathbb{E}_{\theta}\left[\sum_{t=1}^{T}\sum_{i=1}^{d}\left(\frac{L}{\sqrt{d}} - \mathbf{x}_{t,i}\text{sign}(\theta_i)\right)\right] \\
&\geq \frac{\Delta\sqrt{d}}{2L}\mathbb{E}_{\theta}\left[\sum_{t=1}^{T}\sum_{i=1}^{d}\left(\frac{L}{\sqrt{d}} - \mathbf{x}_{t,i}\text{sign}(\theta_i)\right)^2\right] \\
&\geq \frac{\Delta\sqrt{d}}{2L}\sum_{i=1}^{d}\mathbb{E}_{\theta}\left[\sum_{t=1}^{\tau_i}\left(\frac{L}{\sqrt{d}} - \mathbf{x}_{t,i}\text{sign}(\theta_i)\right)^2\right]
\end{aligned}$$

Where the first inequality uses that $\|\mathbf{x}_t\|_2^2 \leq L^2$. Fix $i \in [d]$. For $x \in \{-1, 1\}$, define $U_i(x) = \sum_{t=1}^{\tau_i}\left(\frac{L}{\sqrt{d}} - \mathbf{x}_{t,i}x\right)^2$. And let $\theta' \in \{\pm\Delta\}^d$ be another parameter vector with $\theta_j = \theta'_j$ for $j \neq i$ and $\theta'_i = -\theta_i$. Assume without loss of generality that $\theta_i > 0$. Let $\mathbb{P}_{\boldsymbol{\theta}}$ and $\mathbb{P}_{\boldsymbol{\theta}'}$) be the laws of $U_i(1)$ w.r.t. the bandit learner interaction measure induced by θ and θ' respectively. By a simple calculation we conclude that:

$$\mathrm{KL}(\mathbb{P}_{\boldsymbol{\theta}}, \mathbb{P}_{\boldsymbol{\theta}'}) \leq 2\frac{\Delta^2}{4R^2}\mathbb{E}_\theta\left[\sum_{t=1}^{\tau_i}\mathbf{x}_{t,i}^2\right] \tag{4.35}$$

Also, observe that:

$$U_i(1) = \sum_{t=1}^{\tau_i}\left(L/\sqrt{d} - \mathbf{x}_{t,i}\right)^2 \leq 2L^2\sum_{t=1}^{\tau_i}\frac{1}{d} + 2\sum_{t=1}^{\tau_i}\mathbf{x}_{t,i}^2 \leq \left(\frac{2L^2T + 2\alpha T}{d} + 2L^2\right)$$

Then:

$$\begin{aligned}
\mathbb{E}_\theta[U_i(1)] &\geq \mathbb{E}_{\theta'}[U_i(1)] - \left(\frac{2L^2T + 2\alpha T}{d} + 2L^2\right)\sqrt{\mathrm{KL}(\mathbb{P}_{\boldsymbol{\theta}}, \mathbb{P}_{\boldsymbol{\theta}'})} \\
&\geq \mathbb{E}_{\theta'}[U_i(1)] - \left(\frac{2L^2T + 2\alpha T}{d} + 2L^2\right)\frac{\Delta}{2R}\sqrt{\mathbb{E}_\theta\left[\sum_{t=1}^{\tau_i}\mathbf{x}_{t,i}^2\right]} \\
&\geq \mathbb{E}_{\theta'}[U_i(1)] - \left(\frac{2L^2T + 2\alpha T}{d} + 2L^2\right)\frac{\Delta}{2R}\sqrt{\frac{T\alpha}{d} + L^2} \\
&\geq \mathbb{E}_{\theta'}[U_i(1)] - \left(\frac{2L^2T + 4\alpha T}{d}\right)\frac{\Delta}{2R}\sqrt{\frac{2T\alpha}{d}}
\end{aligned}$$

The last inequality follows by assuming $\alpha n \geq dL^2$, which holds because by assumption $d \geq L^2$ (recall $\alpha = L^2$).

We can then conclude that:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\theta}}\left[U_i(1)\right] + \mathbb{E}_{\boldsymbol{\theta}'}\left[U_i(-1)\right] &\geq \mathbb{E}_{\boldsymbol{\theta}'}\left[U_i(1) + U_i(-1)\right] - \left(\frac{2L^2T + 4\alpha T}{d}\right)\frac{\Delta}{2R}\sqrt{\frac{2T\alpha}{d}} \\
&= 2\mathbb{E}_{\theta'}\left[\frac{\tau_i L^2}{d} + \sum_{t=1}^{\tau_i}\mathbf{x}_{t,i}^2\right] - \left(\frac{2L^2n + 4\alpha T}{d}\right)\frac{\Delta}{2R}\sqrt{\frac{2T\alpha}{d}} \\
&\geq \min\left(\frac{2\alpha T}{d}, \frac{2L^2T}{d}\right) - \left(\frac{2L^2T + 4\alpha T}{d}\right)\frac{\Delta}{2R}\sqrt{\frac{2T\alpha}{d}}
\end{aligned}$$

Therefore using the Randomization Hammer we conclude there must exist a parameter $\theta \in \{\pm\Delta\}^d$ such that $R_{\pi,(\mathcal{X},\boldsymbol{\theta})}(T,1)$ such that:

$$R_{\pi,(\mathcal{X},\boldsymbol{\theta})}(T,1) \geq d\frac{\Delta\sqrt{d}}{2L}\left(\min\left(\frac{2\alpha T}{d}, \frac{2L^2T}{d}\right) - \left(\frac{2L^2T + 4\alpha T}{d}\right)\frac{\Delta}{2R}\sqrt{\frac{2T\alpha}{d}}\right).$$

Let $\alpha = L^2$ and $\Delta = \frac{R\sqrt{d}}{3\sqrt{2T}L}$. In this case:

$$R_{\pi,(\mathcal{X},\boldsymbol{\theta})}(T,1) \geq \frac{Rd}{6\sqrt{2}}\sqrt{T}.$$

The additional constraint on T comes from the fact we must have that $S \geq \|\boldsymbol{\theta}^\star\|_2 = \sqrt{d}\Delta = \frac{Rd}{3\sqrt{2}TL} \implies T \geq \frac{dR}{3\sqrt{2}LS} = \frac{d}{3\sqrt{2}\sqrt{\text{SNR}}}$. □

Misspecification Lower Bound

In this section record a scale-aware version of the lower bound for misspecified linear bandits from [56]. For future reference, recall the definition of misspecification stated in Assumption 3 specialized to a finite context set of size m: Let $\mathcal{X} \subset \mathbb{R}^d$ be a finite context set of size m. A function $f : \mathcal{X} \to \mathbb{R}$ is $\epsilon-$close to linear if there is a parameter $\boldsymbol{\theta}^\star$ such that for all $\mathbf{x} \in \mathcal{X}$,

$$|f(\mathbf{x}) - \mathbf{x}^\top \boldsymbol{\theta}^\star| \leq \epsilon.$$

Before stating our main result we will require the following supporting lemma from [56],

Lemma 15. *For any $\epsilon, L > 0$, and $d \in [m]$ with $d \geq \lceil 8\log(m)L^2/\epsilon^2 \rceil$, and an action set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^d$ satisfying $\|\mathbf{x}_i\|_2 = L$ for all i and such that for all $i, j \in [m]$ with $i \neq j$, $|\mathbf{x}_i^\top \mathbf{x}_j| \leq L^2\sqrt{\frac{8\log(m)}{d-1}}$.*

Proof. This is simply a slightly modified version of in [56, Lemma 3.1]. □

Lemma 16. *Let $\epsilon, L > 0$ and $m \in \mathbb{N}$. For any $d \in [m]$ with $d \geq \lceil 8\log(m)L^2/\epsilon^2 \rceil$ consider a finite action set $\mathcal{X} \subset \mathbb{R}^d$ of size m satisfying $\|\mathbf{x}_i\| = L$ for all $i \in [m]$. Then for any policy π, there is a parameter $\boldsymbol{\theta}^\star$, ϵ-close to a function $f : \mathcal{X} \to \mathbb{R}^m$ for which:*

$$R_{\pi,(\mathcal{X},\boldsymbol{\theta}^\star,f)}(T,1) \geq \epsilon\sqrt{\frac{d-1}{8\log(m)}}\frac{\min(T, m-1)}{4}.$$

Moreover, this parameter $\boldsymbol{\theta}^\star$ satisfies $\|\boldsymbol{\theta}^\star\|_2 = \frac{\epsilon}{L}\sqrt{\frac{d-1}{8\log(m)}}$.

Proof. Observe that by assumption since $d \geq \lceil 8\log(m)L^2/\epsilon^2 \rceil$ we have that $\epsilon \geq L\sqrt{\frac{8\log(m)}{d}}$. By Lemma 15, we may choose $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^d$ satisfying:

1. $\|\mathbf{x}_i\|_2 = L$ for all $i \in [m]$.
2. $|\mathbf{x}_i^\top \mathbf{x}_j| \leq L^2\sqrt{\frac{8\log(m)}{d-1}}$ for all $i \neq j$.

We now consider a family of m bandit instances indexed by each of the arms $\mathbf{x}_i \in \mathcal{X}$. For any $\mathbf{x}_i \in \mathcal{X}$ define $\boldsymbol{\theta}_i^\star = \delta \mathbf{x}_i$ with $\delta = \frac{\epsilon}{L^2}\sqrt{\frac{d-1}{8\log(m)}}$ and define f_i as:

$$f_i(\mathbf{x}) = \begin{cases} L^2\delta & \text{if } \mathbf{x} = \mathbf{x}_i \\ 0 & \text{o.w.} \end{cases}$$

By definition f_i is $\epsilon-$close to linear since for all $\mathbf{x} \in \mathcal{X}$ with $\mathbf{x} \neq \mathbf{x}_i$,

$$\langle \mathbf{x}, \boldsymbol{\theta}_i^\star \rangle \leq \delta L^2 \sqrt{\frac{8\log(m)}{d-1}} = \epsilon \quad \text{and} \quad \langle \mathbf{x}_i, \boldsymbol{\theta}^\star \rangle = L^2\delta$$

Denote by $\mathbf{x}^{(t)}$ the action played by algorithm π at time t and define

$$\tau_i = \max\left\{t \leq n : \mathbf{x}^{(s)} \neq \mathbf{x}_i \forall s \leq t\right\}$$

. Then $\mathbb{E}[R_{\pi,(\mathcal{X},\boldsymbol{\theta}^\star,f)}(T,1)] \geq L^2\delta\mathbb{E}_i[\tau_i]$. Where $\mathbb{E}_i$ denotes the expectation under the law of bandit problem $\boldsymbol{\theta}_i^\star$ and algorithm π. Observe that any algorithm π that queries arm $\mathbf{x}_j$ with $j \neq i$ more than once before pulling arm $\mathbf{x}_i$ will have a larger $\mathbb{E}_i[\tau_i]$ than one that only queries each arm once before querying arm $\mathbf{x}_i$. This means that in order to lower bound $\mathbb{E}_i[\tau_i]$ we can restrict ourselves to algorithms π that do not repeat an arm pull before τ_i. In fact we can assume algorithm π behaves the same for all $i \in [m]$. Observe that for such algorithms whenever facing problem $\boldsymbol{\theta}_i^\star$ and $t \leq \tau_i$, the law of the rewards is independent of $\mathbf{x}_i$ for all $i \in [m]$. Let $f_0 : \mathcal{X} \to \mathbb{R}$ denote the zero function such that $f(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X}$ and let $\mathbb{E}_0$ be the expectation under the law of the bandit problem induced by f_0 and algorithm π. Let T_i be the first time that algorithm A encountered arm i when interacting with f_0. Observe that since the interactions of π with f_i before $\tau_i + 1$ and the interactions of $\mathcal{A}$ with f_0 before $\mathcal{A}$ pulls $\mathbf{x}_i$ (or the time runs up) are indistinguishable $\mathbb{E}_i[\tau_i] = \mathbb{E}_0[\min(T, T_i - 1)]$. A simple averaging argument shows that

$$\begin{aligned}\frac{1}{m}\sum_i R_{\pi,(\mathcal{X},\boldsymbol{\theta}^\star,f)}(T,1) &\geq \frac{L^2\delta}{m}\sum_{i=1}^m \mathbb{E}_i[\tau_i] = \frac{L^2\delta}{m}\sum_{i=1}^m \mathbb{E}_0\left[\min(T, T_i - 1)\right] \\ &= \frac{L^2\delta}{m}\mathbb{E}_0\left[\sum_{i=1}^m \min(T, T_i - 1)\right]\end{aligned}$$

Since π is assumed to interact with f_0 by never pulling the same arm twice, $\{T_i - 1\}_{i=1}^m = \{i-1\}_{i=1}^m$. Using this fact, we can write

$$\frac{1}{m}\sum_{i=1}^m \min(T, T_i - 1) = \frac{1}{m}\sum_{i=1}^{\min(m,T)} i - 1 + \mathbf{1}(T \leq m-1)T.$$

In order to bound the expression above we analyze two cases, first when $T \leq m$, and second when $T > m$. In the first case

$$\frac{1}{m}\sum_{i=1}^{\min(m,T)} i - 1 + \mathbf{1}(T \leq m-1)T = \frac{1}{m}\left(\frac{T(T-1)}{2} + T(m-T).\right)$$

Let's consider two sub-cases. If $T-1 \geq \frac{m}{2}$, then $\frac{1}{m}\left(\frac{T(T-1)}{2} + T(m-T)\right) \geq \frac{T}{4}$. If $T-1 < \frac{m}{2}$ then $m - T > m - \left(\frac{m}{2}+1\right) = \frac{m}{2} - 1$. And therefore $\frac{1}{m}\left(\frac{T(T-1)}{2} + T(m-T)\right) \geq \frac{T}{2} - \frac{T}{m} > \frac{T}{2} - 1$. It follows that whenever $T \leq m$ (and $T > 1$), then $\frac{1}{m}\left(\frac{T(T-1)}{2} + T(m-T)\right) \geq \frac{T}{4}$.

Now let's consider the case when $T > m$. If this holds,

$$\frac{1}{m}\sum_{i=1}^{\min(m,T)} i - 1 + \mathbf{1}(T \leq m-1)T = \frac{m-1}{2}.$$

Assembling these facts together we conclude that in all cases,

$$\frac{1}{m}\sum_{i=1}^{m} \min(T, T_i - 1) \geq \frac{\min(T, m-1)}{4}.$$

The result follows by noting this implies there must exist one $\boldsymbol{\theta}_i^\star$ such that

$$R_{\pi,(\mathcal{X},\boldsymbol{\theta}^\star,f)}(T,1) \geq \epsilon\sqrt{\frac{d-1}{8\log(m)}}\frac{\min(T,m-1)}{4}.$$

The result follows. □

Bibliography

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. "Improved algorithms for linear stochastic bandits". In: (2011), pp. 2312–2320.

[2] Marc Abeille, Alessandro Lazaric, et al. "Linear thompson sampling revisited". In: *Electronic Journal of Statistics* 11.2 (2017), pp. 5165–5197.

[3] Alekh Agarwal et al. "Making contextual decisions with low technical debt". In: *arXiv preprint arXiv:1606.03966* (2016).

[4] Shipra Agrawal and Navin Goyal. "Further optimal regret bounds for thompson sampling". In: (2013), pp. 99–107.

[5] Shipra Agrawal and Navin Goyal. "Thompson sampling for contextual bandits with linear payoffs". In: (2013), pp. 127–135.

[6] Christof Angermueller et al. "Deep learning for computational biology". In: *Molecular systems biology* 12.7 (2016), p. 878.

[7] Christof Angermueller et al. "Population-Based Black-Box Optimization for Biological Sequence Design". In: *arXiv preprint arXiv:2006.03227* (2020).

[8] Frances H Arnold. "Design by directed evolution". In: *Accounts of chemical research* 31.3 (1998), pp. 125–131.

[9] A. Auton, S. Myers, and G. McVean. "Identifying recombination hotspots using population genetic data". In: *arXiv: 1403.4264* (2014).

[10] Ruth E Baker et al. "Mechanistic models versus machine learning, a fight worth fighting for the biological community?" In: *Biology letters* 14.5 (2018), p. 20170660.

[11] Luis A Barrera et al. "Survey of variation in human transcription factors reveals prevalent DNA binding changes". In: *Science* 351.6280 (2016), pp. 1450–1454.

[12] Mohsen Bayati et al. "Unreasonable Effectiveness of Greedy Algorithms in Multi-Armed Bandit with Many Arms". In: *Advances in Neural Information Processing Systems* 33 (2020).

[13] M. A. Beaumont, W. Zhang, and D. J. Balding. "Approximate Bayesian computation in population genetics". In: *Genetics* 162.4 (2002), pp. 2025–2035.

[14] David Belanger et al. "Biological Sequences Design using Batched Bayesian Optimization". In: (2019).

[15] Alberto Bietti, Alekh Agarwal, and John Langford. "A contextual bandit bake-off". In: *arXiv preprint arXiv:1802.04064* (2018).

[16] MGB Blum and O François. "Non-linear regression models for Approximate Bayesian Computation". In: *Statistics and Computing* 20.1 (2010), pp. 63–73.

[17] S. Boitard et al. "Inferring population size history from large samples of genome-wide molecular data-an approximate Bayesian computation approach". In: *PLoS genetics* 12.3 (2016), e1005877.

[18] David H Brookes and Jennifer Listgarten. "Design by adaptive sampling". In: *arXiv preprint arXiv:1810.03714* (2018).

[19] Alon Brutzkus and Amir Globerson. "Globally optimal gradient descent for a convnet with gaussian inputs". In: *arXiv preprint arXiv:1702.07966* (2017).

[20] Jeffrey Chan and Yun S Song. "A Structured Permutation-Equivariant Network for Reference-free Archaic Admixture". In: *NIPS Computational Biology Workshop*. 2017.

[21] Jeffrey Chan et al. "A likelihood-free inference framework for population genetic data using exchangeable neural networks". In: *Advances in neural information processing systems* 31 (2018), p. 8594.

[22] Olivier Chapelle and Lihong Li. "An empirical evaluation of thompson sampling". In: (2011), pp. 2249–2257.

[23] Thomas Desautels, Andreas Krause, and Joel W Burdick. "Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization". In: *Journal of Machine Learning Research* 15 (2014), pp. 3873–3923.

[24] Olga Dolgova and Oscar Lao. "Evolutionary and medical consequences of archaic introgression into modern human genomes". In: *Genes* 9.7 (2018), p. 358.

[25] Miroslav Dudik et al. "Efficient optimal learning for contextual bandits". In: *arXiv preprint arXiv:1106.2369* (2011).

[26] Richard Durrett. *Probability models for DNA sequence evolution*. Springer Science & Business Media, 2008.

[27] P. Fearnhead. "SequenceLDhot: detecting recombination hotspots". In: *Bioinformatics* 22 (24 2006), pp. 3061–3066.

[28] P. Fearnhead and D. Prangle. "Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74.3 (2012), pp. 419–474.

[29] Lex Flagel, Yaniv J Brandvain, and Daniel R Schrider. "The Unreasonable Effectiveness of Convolutional Neural Networks in Population Genetic Inference". In: *bioRxiv* (2018), p. 336073.

[30] Dylan Foster et al. "Practical contextual bandits with regression oracles". In: (2018), pp. 1539–1548.

[31] R. A. Gibbs et al. "The international HapMap project". In: *Nature* 426.6968 (2003), pp. 789–796.

[32] R. C. Griffiths. "Neutral two-locus multiple allele models with recombination". In: *Theoretical Population Biology* 19.2 (1981), pp. 169–186.

[33] C. Guo et al. "On Calibration of Modern Neural Networks". In: *arXiv:1706.04599* (2017).

[34] N. Guttenberg et al. "Permutation-equivariant neural networks applied to dynamics prediction". In: *arXiv:1612.04530* (2016).

[35] Kam Hamidieh. "A data-driven statistical model for predicting the critical temperature of a superconductor". In: *Computational Materials Science* 154 (2018), pp. 346–354.

[36] Michael F Hammer et al. "Genetic evidence for archaic admixture in Africa". In: *Proceedings of the National Academy of Sciences* 108.37 (2011), pp. 15123–15128.

[37] Robert E Hawkins, Stephen J Russell, and Greg Winter. "Selection of phage antibodies by binding affinity: mimicking affinity maturation". In: *Journal of molecular biology* 226.3 (1992), pp. 889–896.

[38] J. Hey. "What's So Hot about Recombination Hotspots?" In: *PLoS Biol* 2.6 (2004), e190.

[39] Eshcar Hillel et al. "Distributed exploration in multi-armed bandits". In: *Advances in Neural Information Processing Systems* 26 (2013), pp. 854–862.

[40] R. R. Hudson. "Properties of a neutral allele model with intragenic recombination". In: *Theoretical population biology* 23.2 (1983), pp. 183–201.

[41] R. R. Hudson. "Two-locus sampling distributions and their application". In: *Genetics* 159.4 (2001), pp. 1805–1817.

[42] P. A. Jenkins and Y. S. Song. "An asymptotic sampling formula for the coalescent with Recombination". In: *The Annals of Applied Probability* 20.3 (2010), pp. 1005–1028.

[43] B. Jiang et al. "Learning Summary Statistic for Approximate Bayesian Computation via Deep Neural Network". In: *arXiv:1510.02175* (2015).

[44] Chi Jin et al. "Minimizing Nonconvex Population Risk from Rough Empirical Risk". In: *arXiv preprint arXiv:1803.09357* (2018).

[45] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. "Online learning under delayed feedback". In: (2013), pp. 1453–1461.

[46] J. A. Kamm et al. "Two-locus likelihoods under variable population size and fine-scale recombination rate estimation". In: *Genetics* 203.3 (2016), pp. 1381–1399.

[47] Kirthevasan Kandasamy et al. "Parallelised bayesian optimisation via thompson sampling". In: (2018), pp. 133–142.

[48] Tarun Kathuria, Amit Deshpande, and Pushmeet Kohli. "Batched gaussian process bandit optimization via determinantal point processes". In: (2016), pp. 4206–4214.

[49] J. Kelleher, A. M. Etheridge, and G. McVean. "Efficient coalescent simulation and genealogical analysis for large sample sizes". In: *PLoS computational biology* 12.5 (2016), e1004842.

[50] D. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv:1412.6980* (2014).

[51] J. F. C. Kingman. "The coalescent". In: *Stochastic processes and their applications* 13.3 (1982), pp. 235–248.

[52] John FC Kingman. "On the genealogy of large populations". In: *Journal of Applied Probability* 19.A (1982), pp. 27–43.

[53] A. Kong et al. "Rate of de novo mutations and the importance of father's age to disease risk". In: *Nature* 488.7412 (2012), pp. 471–475.

[54] Nathan Korda, Balazs Szorenyi, and Shuai Li. "Distributed clustering of linear bandits in peer to peer networks". In: (2016), pp. 1301–1309.

[55] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles". In: (2017), pp. 6402–6413.

[56] Tor Lattimore, Csaba Szepesvari, and Gellert Weisz. "Learning with good feature representations in bandits and in rl with a generative model". In: (2020), pp. 5662–5670.

[57] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[58] Heng Li and Richard Durbin. "Inference of human population history from individual whole-genome sequences". In: *Nature* 475.7357 (2011), pp. 493–496.

[59] J. Li, M. Q. Zhang, and X. Zhang. "A new method for detecting human recombination hotspots and its applications to the HapMap ENCODE data". In: *The American Journal of Human Genetics* 79.4 (2006), pp. 628–639.

[60] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: (2015), pp. 3431–3440.

[61] G. A. T. McVean et al. "The Fine-Scale Structure of Recombination Rate Variation in the Human Genome". In: *Science* 304 (5670 2004), pp. 581–584.

[62] G. Papamakarios and I. Murray. "Fast ϵ-free Inference of Simulation Models with Bayesian Conditional Density Estimation". In: *arXiv:1605.06376* (2016).

[63] P. Pavlidis, J. D. Jensen, and W. Stephan. "Searching for footprints of positive selection in whole-genome SNP data from nonequilibrium populations". In: *Genetics* 185.3 (2010), pp. 907–922.

[64] T. D. Petes. "Meiotic recombination hot spots and cold spots". In: *Nature Reviews Genetics* 2.5 (2001), pp. 360–369.

[65] Vincent Plagnol and Jeffrey D Wall. "Possible ancestral structure in human populations". In: *PLoS genetics* 2.7 (2006), e105.

[66] J K Pritchard et al. "Population growth of human Y chromosomes: a study of Y chromosome microsatellites". In: *Mol Biol Evol* 16.12 (1999), pp. 1791–8.

[67] Kay Prüfer et al. "The complete genome sequence of a Neandertal from the Altai Mountains". In: *Nature* 505.7481 (2014), p. 43.

[68] Ali Rahimi, Benjamin Recht, et al. "Random Features for Large-Scale Kernel Machines." In: 3.4 (2007), p. 5.

[69] S. Ravanbakhsh, J. Schneider, and B. Poczos. "Deep Learning with Sets and Point Clouds". In: *arXiv:1611.04500* (2016).

[70] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. "Deep learning with sets and point clouds". In: *arXiv preprint arXiv:1611.04500* (2016).

[71] Siamak Ravanbakhsh, Jeff Schneider, and Barnabás Póczos. "Equivariance Through Parameter-Sharing". In: Proceedings of Machine Learning Research 70 (June 2017). Ed. by Doina Precup and Yee Whye Teh, pp. 2892–2901. URL: `http://proceedings.mlr.press/v70/ravanbakhsh17a.html`.

[72] Philip A Romero, Andreas Krause, and Frances H Arnold. "Navigating the protein fitness landscape with Gaussian processes". In: *Proceedings of the National Academy of Sciences* 110.3 (2013), E193–E201.

[73] Daniel Russo et al. "A tutorial on thompson sampling". In: *arXiv preprint arXiv:1707.02038* (2017).

[74] Sriram Sankararaman et al. "The combined landscape of Denisovan and Neanderthal ancestry in present-day humans". In: *Current Biology* 26.9 (2016), pp. 1241–1247.

[75] Sriram Sankararaman et al. "The genomic landscape of Neanderthal ancestry in present-day humans". In: *Nature* 507.7492 (2014), pp. 354–357.

[76] D. R. Schrider and A. D. Kern. "Inferring selective constraint from population genomic data suggests recent regulatory turnover in the human brain". In: *Genome biology and evolution* 7.12 (2015), pp. 3511–3528.

[77] Andaine Seguin-Orlando et al. "Genomic structure in Europeans dating back at least 36,200 years". In: *Science* 346.6213 (2014), pp. 1113–1118.

[78] Fathima Aidha Shaikh and Stephen G Withers. "Teaching old enzymes new tricks: engineering and evolution of glycosidases and glycosyl transferases for improved glycoside synthesis". In: *Biochemistry and Cell Biology* 86.2 (2008), pp. 169–177.

[79] S. Sheehan and Y. S. Song. "Deep Learning for Population Genetic Inference". In: *PLoS Computational Biology* 12.3 (2016), e1004845.

[80] Sara Sheehan, Kelley Harris, and Yun S Song. "Estimating variable effective population sizes from multiple genomes: a sequentially Markov conditional sampling distribution approach". In: *Genetics* 194.3 (2013), pp. 647–662.

[81] P. K. Shivaswamy and T. Jebara. "Permutation invariant svms". In: (2006), pp. 817–824.

[82] Sam Sinai and Eric Kelsic. "A primer on model-guided exploration of fitness landscapes for biological sequence design". In: *arXiv preprint arXiv:2010.10614* (2020).

[83] Sam Sinai et al. "AdaLead: A simple and robust adaptive greedy search algorithm for sequence design". In: *arXiv preprint* (2020).

[84] V. C. Sousa et al. "Approximate Bayesian Computation Without Summary Statistics: The Case of Admixture". In: *Genetics* 181.4 (2009), pp. 1507–1519.

[85] Adith Swaminathan and Thorsten Joachims. "Counterfactual risk minimization: Learning from logged bandit feedback". In: (2015), pp. 814–823.

[86] Balazs Szorenyi et al. "Gossip-based distributed stochastic bandit algorithms". In: (2013), pp. 19–27.

[87] Fumio Tajima. "Evolutionary relationship of DNA sequences in finite populations". In: *Genetics* 105.2 (1983), pp. 437–460.

[88] William R Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". In: *Biometrika* 25.3/4 (1933), pp. 285–294.

[89] Joel A Tropp. "User-friendly tail bounds for sums of random matrices". In: *Foundations of computational mathematics* 12.4 (2012), pp. 389–434.

[90] Claire Vernade, Olivier Cappé, and Vianney Perchet. "Stochastic bandit models for delayed conversions". In: *arXiv preprint arXiv:1706.09186* (2017).

[91] Claire Vernade et al. "Linear bandits with stochastic delayed feedback". In: (2020), pp. 9712–9721.

[92] Benjamin Vernot and Joshua M Akey. "Resurrecting surviving Neandertal lineages from modern human genomes". In: *Science* 343.6174 (2014), pp. 1017–1021.

[93] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge University Press, 2019.

[94] J. D. Wall and L. S. Stevison. "Detecting Recombination Hotspots from Patterns of Linkage Disequilibrium". In: *G3: Genes, Genomes, Genetics* (2016).

[95] Y. Wang and B. Rannala. "Population genomic inference of recombination rates and hotspots". In: *Proceedings of the National Academy of Sciences* 106.15 (2009), pp. 6215–6219.

[96] Yuanhao Wang et al. "Distributed bandit learning: Near-optimal regret with efficient communication". In: *arXiv preprint arXiv:1904.06309* (2019).

[97] D. Wegmann, C. Leuenberger, and L. Excoffier. "Efficient Approximate Bayesian Computation coupled with Markov chain Monte Carlo without likelihood". In: *Genetics* 182.4 (2009), pp. 1207–1218.

[98] M. Zaheer et al. "Deep Sets". In: *Neural Information Processing Systems* (2017).

[99] Andrea Zanette et al. "Learning Near Optimal Policies with Low Inherent Bellman Error". In: *arXiv preprint arXiv:2003.00153* (2020).

[100] Zhengyuan Zhou, Renyuan Xu, and Jose Blanchet. "Learning in generalized linear contextual bandits with stochastic delays". In: (2019), pp. 5197–5208.