Automated Machine Learning: New Advances on Bayesian Optimization

Dmitrii Kharkovskii
Specialist Diploma (Hons.), Saint Petersburg State University

A thesis submitted for the degree of
Doctor of Philosophy

Department of Computer Science, School of Computing
National University of Singapore

August 2020

Supervisor:
Associate Professor Low Kian Hsiang

Examiners:
Professor Ng See Kiong
Assistant Professor Jonathan Mark Scarlett
Assistant Professor Ye Nan, University Of Queensland

Automated Machine Learning:
New Advances on Bayesian Optimization

by

Dmitrii Kharkovskii

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Dmitrii Kharkovskii August 17, 2020

Abstract

Recent advances in *Bayesian optimization* (BO) have delivered a promising suite of tools for optimizing an unknown expensive to evaluate black-box objective function with a finite budget of evaluations. A significant advantage of BO is its general formulation: BO can be utilized to optimize any black-box objective function. As a result, BO has been applied in a wide range of applications such as automated machine learning, robotics or environmental monitoring, among others. Furthermore, its general formulation makes BO attractive for deployment in new applications. However, potential new applications can have additional requirements not satisfied by the classical BO setting. In this thesis, we aim to address some of these requirements in order to scale up BO technology for the practical use in new real-world applications.

Firstly, this thesis tackles the problem of data privacy, which is not addressed by the standard setting of BO. Specifically, we consider the outsourced setting where the entity holding the dataset and the entity performing BO are represented by different parties, and the dataset cannot be released non-privately. For example, a hospital holds a dataset of sensitive medical records and outsources the BO task on this dataset to an industrial AI company. We present the *private-outsourced-Gaussian process-upper confidence bound* (PO-GP-UCB) algorithm, which is the first algorithm for privacy-preserving BO in the outsourced setting with a provable performance guarantee. The key idea of our approach is to make the BO performance of our algorithm similar to that of non-private GP-UCB run using the original dataset, which is achieved by using a random projection-based transformation that preserves both privacy and the pairwise distances between inputs. Our main theoretical contribution is to show that a regret bound similar to that of the standard GP-UCB algorithm can be established for our PO-GP-UCB algorithm. We empirically evaluate the performance of our algorithm with synthetic and real-world datasets.

Secondly, we consider applications of BO for hotspot sampling in spatially varying phenomena. For such applications, we exploit the structure of the spatially varying phenomenon in order to increase the BO lookahead and, as a result, improve the performance of the algorithm and make it more suitable for practical use in real-world scenarios. To do this, we present a principled multi-staged Bayesian sequential decision algorithm for nonmyopic adaptive BO that, in particular, exploits macro-actions for scaling up to a further lookahead to match up to a larger available budget. To achieve this, we first generalize GP-UCB to a new acquisition function defined with respect to a nonmyopic adaptive macro-action policy, which, unfortunately, is intractable to be optimized exactly due to an uncountable set of candidate outputs. The key novel contribution of our work here is to show that it is in fact possible to solve for a nonmyopic adaptive ϵ-*Bayes-optimal macro-action BO* (ϵ-Macro-BO) policy given an arbitrary user-specified loss bound ϵ via stochastic sampling in each planning stage which requires only a polynomial number of samples in the length of macro-actions. To perform nonmyopic adaptive BO in real time, we then propose an asymptotically optimal anytime variant of our ϵ-Macro-BO algorithm with a performance guarantee. Empirical evaluation on synthetic and real-world datasets shows that our proposed approach outperforms existing state-of-the-art algorithms.

Finally, this thesis proposes a black-box attack for adversarial machine learning based on BO. Since the dimension of the inputs in adversarial learning is usually too high for applying BO directly, our proposed attack applies dimensionality reduction and searches for an adversarial perturbation in a low-dimensional latent space. The key idea of our approach is to automate both the selection of the latent space dimension and the search of the adversarial perturbation in the selected latent space by using BO. Additionally, we use Bayesian optimal stopping to boost the query efficiency of our attack. Performance evaluation using image classification datasets shows that our proposed method outperforms the state-of-the-art black-box adversarial attacks.

Acknowledgements

I would like to express my sincere and deepest gratitude to my advisor, Associate Professor Low Kian Hsiang for providing continuous support, inspiring guidance and valuable advice during my whole study. I would also like to thank the thesis committee members Professor Ng See Kiong, Assistant Professor Jonathan Scarlett and Assistant Professor Ye Nan for their valuable feedback.

I would like to thank all my labmates for sharing this challenging journey with me. I would like to especially thank Zhongxiang Dai for his valuable feedback and discussions.

I would like to thank all my friends for the happy times they shared with me in all these five years. I would not have done all this work without their support.

I would like to thank my family, for all the love, understanding and encouragement they give me during my study in the opposite part of the world. I would like to thank Chaofan for being there for me.

I acknowledge NUS Centre for Research in Privacy Technologies – N-CRiPT (former Sensor-enhanced Social Media Centre – SeSaMe) for offering me financial support.

This thesis is dedicated to my family.

Their unconditional love and support is invaluable.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Design problems are central to developing complex systems in various domains of technology: in healthcare, doctors and pharmacologists need to design new drugs and tools for treating their patients; in manufacturing, engineers need to design new machines and mechanisms efficiently; in software engineering and computer science, researchers and practitioners need to design advanced programs, algorithms and libraries. All complex systems mentioned above can have dozens of different parameters, e.g., parameters of a treatment protocol in healthcare, parameters of a production line in engineering or parameters of a program configuration in software engineering. These parameters can dramatically impact the behavior of the whole complex system. Therefore, finding their optimal values is a crucial part of the system design strategy. However, the process of parameter selection can be very costly, since it can require restarting the clinical trial of the drug or the production line of a mechanism. As a consequence, many parameters of complex systems similar to those described above are often manually set, resulting in potentially suboptimal system

behavior. Additionally, due to the complex structure of the system, these design objectives cannot be expressed in closed form and their gradient information is not available. Therefore, optimizing the performance of the complex system as a function of its parameters can be formulated as *black-box* optimization – global optimization of functions which are expensive to evaluate and do not have a closed form, analytical description or access to gradient information.

Bayesian optimization (BO) has become an increasingly popular method for optimizing highly complex black-box functions due to its sample efficiency. BO has been applied in a wide range of applications like automated machine learning [Bergstra *et al.*, 2011; Hoffman *et al.*, 2014; Snoek *et al.*, 2012; Swersky *et al.*, 2013; Thornton *et al.*, 2013], robotics [Lizotte *et al.*, 2007; Martinez-Cantin *et al.*, 2007], sensor networks [Garnett *et al.*, 2010], reinforcement learning [Brochu *et al.*, 2010] or synthesis of new materials [Li *et al.*, 2017], among others. BO is a sequential design strategy: It maintains a statistical model of the unknown objective function and uses an acquisition function to repeatedly select an input for evaluating the unknown objective function until the budget is exhausted.

A significant advantage of BO is its general formulation: BO can be utilized to optimize any black-box objective function. As a result, BO algorithms have been used in a wide range of problems, as discussed in the previous paragraph. Furthermore, its general formulation and considerable success make BO technology attractive for deployment in new applications. However, potential new applications can have additional requirements not satisfied by the classical BO setting. In this thesis, we aim to address some of these requirements in order to scale up BO technology for the practical use in new real-world applications. In particular, this thesis aims to address the following limitations of the current BO algorithms in order to facilitate future deployment of BO in new applications:

1. **Privacy-preserving BO in outsourced setting.** In many applications, general-purpose optimization is provided by commercial companies. In recent years, such scenarios of optimization as a service have become increasingly prevalent: SigOpt uses BO as a commercial service for black-box global optimization by providing query access to the users [Dewancker *et al.*, 2016], Google Cloud AutoML offers the optimization of the architectures of neural networks as a cloud service and Microsoft Azure provides tools for tuning the hyperparameters of machine learning models as a service. In all these examples, optimization is performed in the *outsourced* setting, in which the entity holding the dataset (referred to as the *curator* hereafter) and the entity performing optimization (referred to as the *modeler* hereafter) are represented by different parties.

   From the point of view of the entity holding the dataset (the curator), the outsourced setting of optimization naturally has to account for privacy issues. These privacy issues arise due to the widespread use of machine learning (ML) models in applications dealing with sensitive datasets such as health care [Yu *et al.*, 2013], insurance [Chong *et al.*, 2005] and fraud detection [Ngai *et al.*, 2011]. On the other hand, the commercial company (the modeler) is often unwilling to share the details of their proprietary optimization algorithm and its implementation. Therefore, for the case of general-purpose optimization in the outsourced setting, the curator and the modeler are represented by different parties with potentially conflicting interests, as further illustrated with the following examples:

   - A hospital is trying to find out which patients are likely to be readmitted soon based on the result of an expensive medical test [Yu *et al.*, 2013]. Due to cost and time constraints, the hospital (curator) is only able to perform the test for a limited number of patients, and thus outsources the

task of selecting candidate patients for testing to an industrial AI company (modeler). In this case, the inputs to BO are medical records of individual patients and the function to maximize (the output measurement) is the outcome of the medical test for different patients, which is used to assess the possibility of readmission. The hospital is unwilling to release the medical records, while the AI company does not want to share the details of their proprietary algorithm.

- A bank aims to identify the loan applicants with the highest return on investment and outsources the task to a financial AI consultancy. In this case, each input to BO is the data of a single loan applicant and the output measurement to be maximized is the return on investment for different applicants. The bank (curator) is unable to disclose the raw data of the loan applicants due to privacy and security concerns, whereas the AI consultancy (modeler) is unwilling to share the implementation of their selection strategy.
- A real estate agency attempts to locate the cheapest private properties in an urban city. Since evaluating every property requires sending an agent to the corresponding location, the agency (curator) outsources the task of selecting candidate properties for evaluation to an AI consultancy (modeler) to save resources. Each input to BO is a set of features representing a single property and the function to minimize (the output measurement) is the evaluated property price. The agency is unable to disclose the particulars of their customers due to legal implications, while the AI consultancy refuses to share their decision-making algorithm.

A popular way of protecting the privacy of the dataset, as required by the scenarios mentioned above, is to apply the cryptographic framework of *differential*

privacy (DP) [Dwork *et al.*, 2006], which has become the state-of-the-art technique for private data release and has been widely adopted in ML [Sarwate and Chaudhuri, 2013]. However, DP is usually achieved by adding noise, which might negatively affect the performance of the optimization algorithm. Therefore, in order to provide a practical BO solution for outsourced setting, our aim is in designing a BO algorithm, which provides privacy guarantees for the dataset and reliable BO performance.

2. **Nonmyopic BO for hotspot sampling.** Black-box optimization can be applied to many practical problems for hotspot sampling in spatially varying phenomena: environmental sensing (e.g., finding a hotspot of peak phytoplankton abundance [Pennington *et al.*, 2016]), mobile sensor networks (e.g., finding a hotspot of road traffic phenomena [Chen *et al.*, 2015]) or monitoring of the indoor environmental quality (e.g., finding a hotspot of indoor temperature phenomena [Choi *et al.*, 2012]). Obtaining samples in such applications can be prohibitively expensive, which typically results in a limited sampling budget. Therefore, the known convergence rates (i.e., *asymptotic* performance guarantees in the limit) [Bull, 2011; Vazquez and Bect, 2010; Srinivas *et al.*, 2010] of existing myopic BO algorithms are not applicable here, making these algorithms suboptimal in this case. To this end, an optimal BO algorithm could be constructed by performing optimization with respect to the given *finite budget*, thus motivating the need for nonmyopic BO algorithms. However, most of the existing nonmyopic BO algorithms have been empirically demonstrated to be effective and tractable for at most a lookahead of 5 observations, which is usually much smaller than the size of the available budget in practice, and causes these algorithms to behave suboptimally. Scaling up a BO algorithm to a further lookahead to match up to a larger available bud-

get would significantly improve the performance of the algorithm and make it more suitable for practical use in real-world scenarios. In the context of hotspot sampling, a natural way to approach the problem of increasing the lookahead would be by exploiting the structure of the spatially varying phenomenon. The question is, therefore, how the structure and correlation within the spatially varying phenomenon can be utilized in order to increase the BO lookahead.

3. **BO for adversarial learning.** Significant advances in artificial intelligence (AI) in recent years have resulted in the remarkable rate of adoption of AI tools in various industries. Among others, these tools have been used in applications with high security risks such as identity verification [Liu *et al.*, 2018], financial services [Heaton *et al.*, 2017] or autonomous driving [Bojarski *et al.*, 2016]. Naturally, this has raised concerns on whether AI technologies, in particular, deep neural networks are vulnerable to adversarial attacks. In response to these concerns, *adversarial machine learning*, which studies vulnerabilities in machine learning algorithms, has emerged as an important area of research.

   In contrast to the traditional machine learning algorithms, which were originally designed for benign environments, the adversarial learning setting assumes the presence of an *attacker* (adversary). The attacker tries to fool the target machine learning model by querying it with a malicious input. In the context of image classification using deep neural networks, the attacker adds a small perturbation which is imperceptible by humans to an input image with the goal of making the network classify the perturbed image incorrectly. In real-life applications, such attacks can lead to devastating consequences. For instance, the attacker can paste a small, specially crafted patch on a "Stop" road sign. While most humans would not find this patch suspicious, the deep learning based system used by an autonomous vehicle would misclassify the sign with the added

patch as "Speed Limit 45" [Eykholt *et al.*, 2018].

The majority of adversarial attacks proposed in the literature are *white-box*: they assume that the attacker has a full knowledge of the target model architecture [Moosavi-Dezfooli *et al.*, 2016; Kurakin *et al.*, 2016; Carlini and Wagner, 2017; Chen *et al.*, 2018]. However, if the target model is already deployed, the attacker is not aware of the model's implementation and is only able to query it with a malicious input and observe the corresponding output, that is, attempt a *black-box* attack [Papernot *et al.*, 2017b; Tu *et al.*, 2019; Ru *et al.*, 2020]. Furthermore, the attacker has to use a limited number of queries to the target model in order to avoid detection. Therefore, the task of searching for a malicious input in a black-box attack can be framed as a black-box optimization problem under a limited budget. Such problem is exactly the one tackled by BO.

Unfortunately, the dimension of the input images is usually too high for applying BO directly: for example, the dimension of the flattened features of the popular CIFAR-10 dataset is 3072. To improve the query efficiency, existing black-box attacks [Chen *et al.*, 2017; Tu *et al.*, 2019; Ru *et al.*, 2020] usually apply dimensionality reduction techniques and search for an adversarial perturbation in a low-dimensional latent space. While the authors of the existing works emphasize the importance of dimensionality reduction on performance of their attacks, they either treat the dimension of the latent space as a hyperparameter [Ru *et al.*, 2020] or set it manually [Chen *et al.*, 2017; Tu *et al.*, 2019]. Providing a principled way for selecting the dimension of the latent space could increase the attack success rate under a limited query budget. The challenge, therefore, is in designing a BO algorithm for performing a black-box adversarial attack, which automatically selects the dimension of the

latent space while being query efficient.

1.2 Objective

The main focus of this thesis is to address the following question:

How can BO be scaled up to satisfy the additional requirements of new real-world applications?

Specifically, the works in this thesis attempt to scale up BO to the following new real-world applications by addressing their corresponding additional requirements:

- **Privacy-preserving BO in outsourced setting.** *How do we design a BO algorithm in the outsourced setting where the entity holding the dataset and the entity performing BO are represented by different parties, and the dataset cannot be released non-privately? Is it possible to ensure the privacy protection of the dataset and, at the same time, obtain theoretical performance guarantees and empirical effectiveness for such an algorithm?*

- **Nonmyopic BO for hotspot sampling.** *How can the structure of the spatially varying phenomenon be exploited for scaling up a BO algorithm to a further lookahead to match up to a larger available budget in hotspot sampling applications? Is it possible to achieve optimal expected performance with respect to the given finite budget?*

- **BO for adversarial learning.** *In order to design a practical BO algorithm for performing a black-box adversarial attack, how can the dimension of the latent space be selected in a principled way? How can the designed algorithm ensure the query efficiency?*

The questions mentioned above are then considered and tackled in this thesis as described next.

1.3 Contributions

With regard to the objectives introduced in the previous section, the works in this thesis support the following statements:

- *It is possible to construct a privacy-preserving and empirically effective Bayesian optimization algorithm in outsourced setting. The constructed algorithm protects the privacy of the dataset using differential privacy, fulfills theoretical performance guarantees and shows the empirical effectiveness.*

- *Exploiting macro-actions can scale a Bayesian optimization algorithm up to a further lookahead to match up to a larger available budget. For a given finite budget, it is possible to guarantee an ϵ-Bayes-optimal expected performance for such an algorithm with respect to an arbitrary, user-defined loss bound ϵ, by using the notion of Bayes-optimality.*

- *Selection of the dimension of the latent space can be automated using Bayesian optimization. Bayesian optimal stopping can be used in order to preserve the query efficiency of the designed algorithm.*

These claims are substantiated by the following novel contributions:

1. **Private Outsourced BO (Chapter 3)**.

   We propose the *private-outsourced-Gaussian process-upper confidence bound* (PO-GP-UCB) algorithm, which is the first algorithm for BO with differential privacy in the outsourced setting with a provable performance guarantee. The key idea of our approach is to make the Gaussian Process (GP) predictions

and hence the BO performance of our algorithm similar to those of non-private GP-UCB [Srinivas *et al.*, 2010] run using the original dataset. To achieve this, instead of standard differential privacy methods, we use a privacy-preserving transformation based on random projection [Johnson and Lindenstrauss, 1984], which approximately preserves the pairwise distances between inputs. We show that preserving the pairwise distances between inputs leads to preservation of the GP predictions and therefore the BO performance in the outsourced setting (compared with the standard setting of running non-private GP-UCB [Srinivas *et al.*, 2010] on the original dataset). Our main theoretical contribution is to show that a regret bound similar to that of the standard GP-UCB algorithm can be established for our PO-GP-UCB algorithm. We empirically evaluate the performance of our algorithm with synthetic and real-world datasets.

2. **Nonmyopic BO with macro-actions (Chapter 4)**.

   We present a principled multi-staged Bayesian sequential decision algorithm for nonmyopic adaptive BO that, in particular, exploits macro-actions inherent to the structure of several real-world task environments/applications for scaling up to a further lookahead (as compared to the existing nonmyopic adaptive BO algorithms [Lam *et al.*, 2016; Lam and Willcox, 2017; Ling *et al.*, 2016; Marchant *et al.*, 2014; Osborne *et al.*, 2009]) to match up to a larger available budget. To achieve this, we first generalize GP-UCB [Srinivas *et al.*, 2010] to a new acquisition function defined with respect to a nonmyopic adaptive macro-action policy, which, unfortunately, is intractable to be optimized exactly due to an uncountable set of candidate outputs. The key novel contribution of our work here is to show that it is in fact possible to solve for a nonmyopic adaptive ϵ-*Bayes-optimal macro-action BO* (ϵ-Macro-BO) policy given an arbitrary user-specified loss bound ϵ via stochastic sampling in each planning stage which

requires only a polynomial number of samples in the length of macro-actions. To perform nonmyopic adaptive BO in real time, we then propose an asymptotically optimal anytime variant of our ϵ-Macro-BO algorithm with a performance guarantee. We empirically evaluate the performance of our ϵ-Macro-BO algorithm and its anytime variant in BO with synthetic and real-world datasets.

3. **Black-box adversarial attack automated with BO (Chapter 5)**.
   We propose a novel *Bayesian-Optimization-with-dimension-selection-and-Bayesian-optimal-stopping* (BOS^2) black-box adversarial attack. The key idea of our approach is to increase the attack success rate by using BO for automating both the selection of the latent space dimension and the search of the adversarial perturbation in the selected latent space. Our attack consists of two stages. In the first stage we use BO to select the dimension of the latent space for projecting the high-dimensional input space into (the dimension BO loop). In the second stage we use Add-GP-UCB algorithm [Kandasamy *et al.*, 2015] to search for the adversarial perturbation in the latent space (the perturbation BO loop). To boost the query efficiency of our BOS^2 attack, we use Bayesian optimal stopping [Dai *et al.*, 2019] to early-stop the execution of the perturbation BO loop for those latent dimensions, which will end up under-performing, hence eliminating unnecessary queries. We evaluate the performance of our BOS^2 attack using MNIST and CIFAR-10 datasets to show that our method outperforms the existing black-box adversarial attacks.

1.4 Organization

The remaining chapters of this thesis are organized as follows. Section 2.1 briefly describes the problem setting of BO. The related works are discussed in Section 2.2

(privacy-preserving BO), Section 2.3 (nonmyopic BO) and Section 2.4 (adversarial attacks). Chapter 3 presents our privacy-preserving BO algorithm for outsourced setting PO-GP-UCB. Our nonmyopic BO algorithm for hotspot sampling in spatially varying phenomena is reported in Chapter 4. The BOS^2 black-box adversarial attack using BO is proposed in Chapter 5. Finally, the conclusion and the future works of this thesis are presented in Chapter 6.

Chapter 2

Related Works

2.1 Bayesian Optimization

Bayesian optimization (BO) has become an increasingly popular method for optimizing highly complex black-box functions with expensive function evaluations. Such optimization problems frequently appear in a wide range of applications like automated machine learning [Bergstra *et al.*, 2011; Hoffman *et al.*, 2014; Snoek *et al.*, 2012; Swersky *et al.*, 2013; Thornton *et al.*, 2013], robotics [Lizotte *et al.*, 2007; Martinez-Cantin *et al.*, 2007], sensor networks [Garnett *et al.*, 2010], reinforcement learning [Brochu *et al.*, 2010], among others.

Traditional optimization methods are not applicable for such optimization problems, because the objective function does not have analytical expression or access to gradient information and is expensive to evaluate. The most straightforward way to approach the optimization problem would be using *grid search*, which exhaustively explores the grid of candidate parameter values until a reasonable performance has been reached. However, such an approach is not scalable in terms of the number of parameters. A slightly more advanced *random search*, instead of using a grid,

randomly samples the candidate parameter values. While this approach empirically outperforms its deterministic grid counterpart, it is still time-consuming and poorly scalable to large dimensions. Another approach would be using evolutionary algorithms, however, they are non-deterministic, sensitive to the choice of initialization, and are easily stuck in a local optima[1].

Conventionally, a BO algorithm models the unknown objective function with a *Gaussian Process* (GP) and uses a heuristic called *acquisition function* (AF) to guide the algorithm's search for the global maximum. Specifically, a BO algorithm exploits the chosen AF to repeatedly select an input for evaluating the unknown objective function that trades off between observing a likely maximum based on a GP belief of the unknown objective function (exploitation) vs. improving the GP belief (exploration). After obtaining the candidate input recommended by the AF, a BO algorithm evaluates the objective function at this input, and updates the GP model using the newly obtained value of the objective function. The whole procedure[2] is repeated until the budget is expended. Therefore, in contrast to the grid and random search strategies, a BO algorithm is able to learn from the whole history of past data/observations, resulting in a better performance.

The following sections of this chapter discuss the existing works related to privacy-preserving BO (Section 2.2), nonmyopic BO (Section 2.3) and BO for adversarial attacks on machine learning models (Section 2.4).

[1] In certain cases evolutionary strategies perform better than BO. For example, [Mori *et al.*, 2005] showed that evolutionary strategies outperformed BO on an number of simulated problems. However, a significant number of novel BO algorithms have been proposed after this paper had been published. On the other hand, in the context of black-box adversarial machine learning (Chapter 5), both BO-based attacks (BayesOpt attack [Ru *et al.*, 2020] and our proposed BOS^2 attack) outperform GenAttack [Alzantot *et al.*, 2019] based on evolutionary strategies.

[2] See Algorithm 1 in Section 3.1 for illustration of the procedure of the general BO algorithm.

2.2 Privacy-preserving Bayesian Optimization

The number of prior works on privacy-preserving BO is limited. A work by Kusner *et al.* [2015] proposed a DP variant of GP-UCB algorithm [Srinivas *et al.*, 2010]. The authors consider the task of hyperparameter tuning for machine learning models and introduce methods for privatizing the best hyperparameter configuration and classification accuracy (i.e., best input and output measurement found by GP-UCB) by computing their respective non-private values and releasing them using standard DP mechanisms (Section 3.4). However, such an approach implies that the entity holding the data (the curator) and the entity performing BO (the modeler) are represented by the same party and thus both entities have full access to the sensitive dataset and detailed knowledge of the BO algorithm. In our outsourced setting, in contrast, the modeler only has access to the transformed privatized dataset, while the curator is unaware of the details of the BO algorithm, as described in our motivating scenarios (Section 1.1). Furthermore, in contrast to our work, Kusner *et al.* [2015] do not provide any regret bound and the approximated quality of their computed privatized estimates of the best hyperparameters and classifier accuracy proposed by the authors, in fact, degrades when the number of BO budget increases.

A recent work of Nguyen *et al.* [2018] considers a setting, which resembles that of ours. However, the authors use a self-proposed notion of privacy instead of the widely recognized DP. Furthermore, Nguyen *et al.* [2018] protect the privacy of only the output measurements (in our case, that would be, for example, the outcome of the medical test for the patient or the return on investment for the loan applicant). In contrast, we aim at preserving the privacy of the inputs: For instance, if the input is a medical record, releasing it may unveil the identity of the patient, while releasing only the outcome of the medical test (the output measurement) would not. Similarly, releasing the raw data of the loan applicant may unveil her identity, as opposed to

releasing only the value of the return on investment. Note that, both Kusner *et al.* [2015] and Nguyen *et al.* [2018] do not provide any provable performance bound, while the performance of our algorithm (Chapter 3) is theoretically guaranteed.

2.3 Nonmyopic Bayesian Optimization

In contrast to the existing BO algorithms, our proposed nonmyopic adaptive algorithm ϵ-Macro-BO scales up to a further lookahead (as compared to the existing nonmyopic adaptive BO algorithms) and provides a theoretical guarantee for the expected performance loss. These characteristics distinguish our algorithm from the existing works and are discussed in greater detail with the related work below.

2.3.1 Single-point vs. batch algorithms

Conventionally, a BO algorithm exploits the chosen acquisition function to repeatedly select an input for evaluating the unknown objective function until the budget is expended. Such an algorithm selects one candidate input at a time, hence we call it *single-point*. A number of myopic (nonmyopic approaches will be discussed later in Section 2.3.2) single-point algorithms were proposed in the literature, including *probability of improvement* (PI) or *expected improvement* (EI) over currently found maximum [Shahriari *et al.*, 2016], information-based [Hennig and Schuler, 2012; Hernández-Lobato *et al.*, 2014; Villemonteix *et al.*, 2009], or upper confidence bound (UCB) [Srinivas *et al.*, 2010].

Unfortunately, such a conventional BO algorithm is greedy/myopic and hence performs suboptimally with respect to the given finite budget: While acquisition functions like EI [Bull, 2011; Vazquez and Bect, 2010] and UCB [Srinivas *et al.*, 2010] offer theoretical guarantees for the convergence rate of their BO algorithms

(i.e., in the limit) via regret bounds, in practice, since the budget is limited, such bounds are suboptimal as they cannot be specified to be arbitrarily small. In contrast, the performance loss of our proposed algorithm ϵ-Macro-BO can be bounded in the expected sense by an arbitrary user-specified value for a given BO budget.

To be nonmyopic, the BO algorithm's policy to select the next input has to additionally account for its subsequent selections of inputs for evaluating the unknown objective function. Perhaps surprisingly, this can be partially achieved by *batch* BO algorithms [Azimi *et al.*, 2010; Contal *et al.*, 2013; Desautels *et al.*, 2014; González *et al.*, 2016a; Chevalier and Ginsbourger, 2013; Daxberger and Low, 2017; Shah and Ghahramani, 2015; Wu and Frazier, 2016]. In contrast to conventional BO algorithms described above, batch BO algorithms repeatedly select a set of multiple inputs for querying the objective in parallel at every iteration – a *batch*. Batch BO algorithms can be classified into two types. Greedy batch BO algorithms [Azimi *et al.*, 2010; Contal *et al.*, 2013; Desautels *et al.*, 2014; González *et al.*, 2016a] select the inputs of a batch one at a time in a greedy manner and hence are myopic. In contrast, others [Chevalier and Ginsbourger, 2013; Daxberger and Low, 2017; Shah and Ghahramani, 2015; Wu and Frazier, 2016] are capable of *jointly* optimizing a batch of inputs because their selection of each input has to account for that of all other inputs of the batch[3]. However, since the batch size is typically set to be much smaller than the given budget, batch BO algorithms have to repeatedly select the next batch greedily. Furthermore, unlike the conventional BO algorithms described above, their selection of each input is independent of the outputs observed from evaluating the objective function at the other selected inputs of the batch, thus sacrificing some degree of adaptivity. Hence, batch BO algorithms also perform suboptimally with respect to the given budget. In contrast, our proposed algorithm ϵ-Macro-BO

[3]Batch BO is traditionally considered when resources are available to evaluate the objective function in parallel. We deviate from such a tradition here and suggest a further possibility of using batch BO for nonmyopic selection of inputs.

is fully adaptive and selects the next macro-action (i.e., the next batch of inputs) in an adaptive nonmyopic manner, thus resolving both the drawbacks of batch BO algorithms described above.

2.3.2 Myopic vs. nonmyopic algorithms

Some nonmyopic adaptive BO algorithms [Lam and Willcox, 2017; Lam *et al.*, 2016; Ling *et al.*, 2016; Marchant *et al.*, 2014; Osborne *et al.*, 2009] have been recently developed to resolve the drawbacks of conventional greedy/myopic BO algorithms described in Section 2.3.1. But, they have been empirically demonstrated to be effective and tractable for at most a lookahead of 5 observations which is usually much less than the size of the available budget in practice, thus causing them to behave myopically in this case. To increase the lookahead, the work of González *et al.* [2016b] has proposed a two-staged approach that utilizes a *greedy* batch BO algorithm in its second stage to efficiently but myopically optimize all but the first input afforded by the budget. Note that the above works on nonmyopic adaptive BO do not provide theoretical performance guarantees except for that of Ling *et al.* [2016]. In contrast, our approach can empirically scale to a lookahead of up to 20 observations (and hence match up to a larger budget) and is still amenable to a theoretical analysis of its performance.

2.3.3 Adaptive vs. non-adaptive algorithms

Adaptive algorithms exploit the outputs observed from evaluating the objective function at the previously selected inputs for selecting the new input. In contrast, non-adaptive algorithms do not use the past history of observations, and hence, the new inputs to be selected can be determined prior to execution of the algorithm. Adaptive algorithms usually outperform the non-adaptive ones, so most of

the single-point myopic algorithms [Hennig and Schuler, 2012; Srinivas *et al.*, 2010; Hernández-Lobato *et al.*, 2014; Villemonteix *et al.*, 2009] are adaptive. As pointed out in Section 2.3.1, batch BO algorithms sacrifice some adaptivity, because the inputs within a batch are independent of the observations within this batch. Designing an adaptive nonmyopic BO algorithm is very challenging and computationally involved, so some proposed nonmyopic BO algorithms are non-adaptive [Marchant *et al.*, 2014]. The existing adaptive algorithms [Lam and Willcox, 2017; Lam *et al.*, 2016; Ling *et al.*, 2016; Marchant *et al.*, 2014; Osborne *et al.*, 2009] are able to scale to the lookahead of only up to 5 observations, which is usually much smaller than the sampling budget for BO tasks. In contrast, our proposed algorithm is adaptive and scales to a larger lookahead of up to 20 observations, resulting in its superior BO performance, as compared to existing works.

2.4 Adversarial attacks on machine learning models

In contrast to the existing adversarial attacks, our proposed BOS^2 attack is a black-box evasion attack, which applies BO to automate both the selection of the latent space dimension and the search of the adversarial perturbation, and uses Bayesian optimal stopping [Dai *et al.*, 2019] in order to boost its query efficiency. We discuss these characteristics, which distinguish our algorithm from the prior works, in the following section.

2.4.1 Poisoning vs. evasion attacks

The two most popular adversarial attack types are *poisoning* attacks and *evasion* attacks. To perform a poisoning attack, the adversary injects the malicious data in the training set of the model, aiming to degrade the performance of the model during

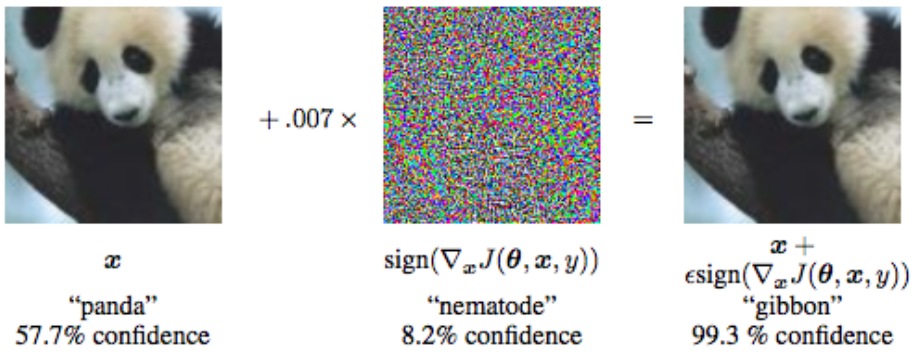


Figure 2.1: Visual illustration of an evasion attack. Image courtesy of [Goodfellow *et al.*, 2014].

testing or deployment, i.e., the adversary is "poisoning" the model [Xiao *et al.*, 2012; Newell *et al.*, 2014; Mei and Zhu, 2015; Koh and Liang, 2017; Feng *et al.*, 2019; Xiao *et al.*, 2015; Burkard and Lagesse, 2017]. On the other hand, in an evasion attack setting, the adversary is attacking a model which is already trained. In this case, the attacker is querying the model with a malicious input. This malicious input is formed by adding a small perturbation to a benign input (which would be classified correctly), such that the model would output an incorrect answer for the perturbed malicious input with high confidence. This setting is visually illustrated in a famous image (Fig. 2.1) from the work of Goodfellow *et al.* [2014]. Since our BOS^2 attack is an evasion attack, we describe the existing attacks of this category in the next few subsections.

2.4.2 Targeted vs. untargeted attacks

The aim of an *untargeted* attack [Kwon *et al.*, 2018; Wu *et al.*, 2019; Suya *et al.*, 2017] is to cause the model to provide the erroneous output (e.g., misclassify the input adversarial image). *Targeted* attacks [Chen *et al.*, 2017; Tu *et al.*, 2019; Ru *et al.*, 2020], in addition to that, aim not only to misclassify the input, but to make the

model predict a given target class. Targeted attacks give the adversary more control over the attacked model, since she can force the model to predict the required target class. Our BOS^2 attack is a targeted attack.

2.4.3 White-box vs. black-box attacks

The majority of evasion attacks proposed in the literature are *white-box*: they assume that the attacker has a full knowledge of the model's architecture and implementation [Goodfellow *et al.*, 2014; Gu and Rigazio, 2014; Moosavi-Dezfooli *et al.*, 2016; Kurakin *et al.*, 2016; Carlini and Wagner, 2017; Chen *et al.*, 2018]. In this case, the attacker, for instance, can obtain the exact gradients of the model and perform back-propagation in order to search for a successful adversarial perturbation. However, typically (e.g., if the machine learning model is already deployed), the attacker would have access only to the output measurements for a given input, which results in a *black-box* attack. Black-box attacks are much more challenging due to very limited information available to the attacker. Since our BOS^2 attack falls in the latter category, existing black-box attacks are discussed in more detail in the next subsection.

2.4.4 Black-box attacks

One class of black-box attacks uses the outputs obtained from the attacked model to train a substitute machine learning model [Papernot *et al.*, 2016]. The adversary then can attack the substitute model in an easier white-box setting and use the successful adversarial example for the substitute model to attack the original model. However, to train a substitute model with similar properties as the original model, the attacker either needs to have access to the training data or to generate a synthetic training dataset by excessively querying the original model. This restriction about training data makes such attacks infeasible for data-intensive domains (e.g., ImageNet

dataset), as pointed out in a number of existing works [Brendel *et al.*, 2017; Liu *et al.*, 2016].

Another approach to black-box attacks is to estimate gradients using zeroth-order optimization (ZOO) [Chen *et al.*, 2017] and then use this estimate to attack the model in a white-box setting. However, the method of Chen *et al.* [2017] requires a large number of queries to the attacked model in order to estimate the gradient accurately and, as a result, is very computationally involved. AutoZOOM [Tu *et al.*, 2019] improves the query efficiency of ZOO attack by considering random vectors to estimate model gradients. While the authors of AutoZOOM emphasize the importance of dimensionality reduction on the performance of their method, they set the dimension of the latent space manually. In contrast, our BOS^2 attack provides a principled way to select the latent dimension using BO. Another work by Ilyas *et al.* [2018] uses natural evolution strategy in order to estimate model gradients.

In addition to the attacks mentioned above, a number of other approaches were proposed. As an example, Brendel *et al.* [2017] introduced the Boundary Attack, which starts from a huge adversarial perturbation (hence, resulting in the misclassification of the perturbed image) and then gradually reduces the perturbation using random walks along the decision boundary. However, this attack has a very high computational complexity due to a very large number of queries required to reduce the distortion. Different from Brendel *et al.* [2017], GenAttack introduced by Alzantot *et al.* [2019] uses genetic algorithms to iteratively evolve the population of candidate adversarial examples.

2.4.5 BO for adversarial attacks

There are a few existing works using BO for adversarial attacks. The earlier work of Suya *et al.* [2017] uses a simple BO method to propose an untargeted attack

on low-dimensional spam email dataset with 57 input features. This method is not scalable to attacks on images with thousands of dimensions, which is the most popular application of adversarial learning. Another work is that of Zhao *et al.* [2019]. The authors apply BO directly on the original high-dimensional input space, resulting in a large distortion of found adversarial examples, since BO is known to work poorly in large dimensions. To make the problem more amenable to BO, our BOS^2 attack uses dimensionality reduction and searches for an adversarial perturbation in the latent space, resulting in smaller distortion.

The closest related work to ours is the recent BayesOpt attack by Ru *et al.* [2020]. The authors use bilinear interpolation to project the original input space into a latent space of lower dimension and then search for an adversarial perturbation in this latent space. However, they treat the latent dimension as a hyperparameter and optimize it without considering the BO procedure. In contrast, our BOS^2 attack learns the optimal latent dimension with BO. To do this, our attack exploits the results of the BO performances from the previously observed latent dimensions. Furthermore, Ru *et al.* [2020] update the latent dimension after a fixed number of queries to the attacked model. As a result, if the BO procedure in the selected latent space is underperforming, the attack of Ru *et al.* [2020] would keep sending unnecessary queries to the model till the next update. In contrast, our BOS^2 attack uses Bayesian optimal stopping [Dai *et al.*, 2019] to early-stop the execution of the perturbation BO loop if the BO performance for the current dimension is unsatisfactory and, hence, avoids redundant queries to the model.

Chapter 3

Private Outsourced Bayesian Optimization

This chapter of the thesis proposes the *private-outsourced-Gaussian process-upper confidence bound* (PO-GP-UCB) algorithm. To the best of our knowledge, our algorithm is the first algorithm for BO with differential privacy (DP) in the outsourced setting with a provable performance guarantee.

To recall, in the outsourced setting the curator is unable to release the original dataset due to privacy concerns, and therefore has to provide a transformed privatized dataset to the modeler. Then, the modeler can perform BO (specifically, the GP-UCB algorithm) on the transformed dataset. A natural choice for the privacy-preserving transformation is to apply standard DP methods such as the Laplace or Gaussian mechanisms [Dwork and Roth, 2014] directly to the original dataset. However, the theoretically guaranteed convergence of the GP-UCB algorithm [Srinivas *et al.*, 2010] is only valid if it is run using the original dataset. Therefore, as a result of the privacy-preserving transformation required in the outsourced setting, it is unclear whether the theoretical guarantee of GP-UCB can be preserved and thus whether

reliable performance can be delivered. To resolve this complication, we follow a different approach: our main idea is to make the GP predictions (and hence the BO performance) of our algorithm similar to those of non-private GP-UCB run using the original dataset. To achieve this, we design a privacy-preserving transformation based on random projection [Johnson and Lindenstrauss, 1984], instead of using the standard DP methods. Such a transformation approximately preserves the pairwise distances between inputs. We show that preserving the pairwise distances between inputs results in preservation of the GP predictions and therefore the BO performance in the outsourced setting (compared with the standard setting of running non-private GP-UCB on the original dataset). We prove that a regret bound similar to that of the standard GP-UCB algorithm can be established for our PO-GP-UCB algorithm, which is our key theoretical contribution.

The rest of this chapter is organized as follows. Some background about BO and GP is stated in Section 3.1. The celebrated GP-UCB algorithm [Srinivas *et al.*, 2010] is summarized in Section 3.2. The problem setting of outsourced BO is introduced in Section 3.3. The framework of differential privacy is reviewed in Section 3.4. Our PO-GP-UCB algorithm, its theoretical performance guarantee and analysis are described in Section 3.5. Experiments using synthetic and real-world datasets for empirical performance evaluation of our PO-GP-UCB algorithm are presented in Section 3.6.

3.1 Background

3.1.1 Formal problem statement of Bayesian Optimization

Bayesian Optimization is tackling the problem of sequentially maximizing an unknown objective function $f : \mathcal{X} \to \mathbb{R}$, in which $\mathcal{X} \subset \mathbb{R}^d$ denotes a domain of d-dimensional inputs:

$$\max_{x \in \mathcal{X}} f(x).$$

We assume that f is an unknown (possibly noisy, non-convex, and/or with no closed-form expression/derivative) objective function, which is very expensive to evaluate, such that only a small number of function evaluations can be made. Furthermore, function f is accessible through the noisy output measurements

$$y(x) \triangleq f(x) + \epsilon_{GP}, \tag{3.1}$$

in which $\epsilon_{GP} \sim \mathcal{N}(0, \sigma_n^2)$ is a zero-mean Gaussian noise with noise variance σ_n^2.

Conventionally, a BO algorithm consists of two major components: the model of the unknown objective function and the *acquisition function* (AF). The model is typically represented by a *Gaussian Process* (GP) (see Section 3.1.2 for details about GP). The AF serves as a heuristic to guide the algorithm's search for the global maximum of the objective function. Specifically, the BO algorithm exploits the chosen AF to repeatedly select an input for evaluating the unknown objective function that trades off between observing a likely maximum based on a GP belief of the unknown objective function (exploitation) vs. improving the GP belief (exploration) until the budget is expended. In each iteration $t = 1, \ldots, T$, an unobserved input $x_t \in \mathcal{X}$ is selected to query the unknown objective function by maximizing the AF, yielding a noisy output measurement $y_t \triangleq f(x_t) + \epsilon_{GP}$, in which $\epsilon_{GP} \sim \mathcal{N}(0, \sigma_n^2)$ is a zero-mean Gaussian noise with noise variance σ_n^2, as defined in (3.1). This procedure is illustrated in Algorithm 1 below.

The AF should be designed to allow the BO algorithm to approach the global maximum $f(x^*)$ rapidly, in which $x^* \triangleq \text{argmax}_{x \in \mathcal{X}} f(x)$. This can be achieved by minimizing a standard BO objective such as *regret*. The notion of regret intuitively refers to a loss in reward resulting from not knowing x^* beforehand. Formally, the

Algorithm 1 General BO algorithm

1: **Input:** Input domain $\mathcal{X}$, sampling budget T, acquisition function α
2: **for** $t = 1, \ldots, T$ **do**
3: Select new input x_t by optimizing the acquisition function α using the currently available data $\mathcal{D}_{t-1}$: $x_t \leftarrow \text{argmax}_x \, \alpha(x, \mathcal{D}_{t-1})$
4: Query the objective function for the noisy output measurement y_t
5: Augment data $\mathcal{D}_t \leftarrow \{\mathcal{D}_{t-1}, (x_t, y_t)\}$
6: Update the GP model
7: **return** the largest found value $\max_{t=1,\ldots,T} y_t$

instantaneous regret incurred in iteration t is defined as

$$r_t \triangleq f(x^*) - f(x_t).$$

Cumulative regret is defined as the sum of all instantaneous regrets, i.e.,

$$R_T \triangleq \sum_{t=1}^{T} r_t,$$

and *simple regret* is defined as the minimum among all instantaneous regrets, i.e.,

$$S_T \triangleq \min_{t=1,\ldots,T} r_t.$$

It is desirable for a BO algorithm to achieve *no regret* asymptotically, i.e.,

$$\lim_{T\to\infty} S_T \leq \lim_{T\to\infty} R_T/T = 0,$$

which implies that it will eventually converge to the global maximum, since the currently found maximum after T iterations is no further away from $f(x^*)$ than R_T/T.

3.1.2 Gaussian Process (GP)

In order to facilitate the design of the AF to minimize the regret, we model our belief of the unknown objective function f using a Gaussian Process (GP) [Rasmussen and Williams, 2006]. Let $f(x)_{x\in\mathcal{X}}$ denote a GP, which is formally defined below:

Definition 3.1. *A Gaussian process (GP) is a collection of random variables, any finite number of which have joint Gaussian distributions [Quiñonero-Candela and Rasmussen, 2005].*

Then, the GP is fully specified by its *prior* mean $\mu_x \triangleq \mathbb{E}[f(x)]$ and covariance function (kernel) $k_{xx'} \triangleq \text{cov}[f(x), f(x')]$ for all $x, x' \in \mathcal{X}$. Without loss of generality, we assume $\mu_x = 0$ for every $x \in \mathcal{X}$. Next we discuss the common choices of covariance function $k_{xx'}$ in Section 3.1.3, followed by the description of GP regression in Section 3.1.4.

3.1.3 Common choices of covariance function

The common choices of covariance function (which we also call *kernels*) $k_{xx'}$ are:

1. The *linear* kernel is defined as

$$k_{xx'} \triangleq x^\top x'.$$

   Using GP with a linear kernel is a special case of Bayesian linear regression.

2. The non-isotropic *squared exponential* (SE) kernel is defined as

$$k_{xx'} \triangleq \sigma_y^2 \cdot e^{(x-x')^\top \Gamma^{-2}(x-x')},$$

   in which Γ is a diagonal matrix with length-scale components $[l_1, \ldots, l_d]$ controlling the correlation or "similarity" between output measurements and σ_y^2 is

the signal variance controlling the intensity of output measurements. When all length-scale components are equal to l, the non-isotropic SE kernel reduces to the *isotropic* SE kernel defined as

$$k_{xx'} \triangleq \sigma_y^2 \cdot e^{-0.5\|x-x'\|^2/l^2}.$$

Note that any non-isotropic SE kernel can be easily transformed to an isotropic one by preprocessing the inputs, i.e., dividing each dimension of inputs x, x' by the respective length-scale component l_i.

3. The *Matérn* kernel is given by

$$k(x, x') \triangleq \left(2^{1-\nu}/\Gamma(\nu)\right) \cdot r^\nu B_\nu(r)$$

where $r \triangleq (\sqrt{2\nu}/l)\|x - x'\|$. Parameters ν, l control the correlation or "similarity" between output measurements and B_ν is the modified Bessel function of the second kind. Note that as $\nu \to \infty$, appropriately rescaled Matérn kernel converges to the isotropic SE kernel.

3.1.4 Gaussian Process regression

Given a set $\mathbf{x}_{1:t} \triangleq \{x_1, \ldots, x_t\}$ of inputs after t iterations and a column vector $\mathbf{y}_{1:t} \triangleq [y_i]^\top_{1,\ldots,t}$ of their corresponding noisy output measurements, a GP model can perform probabilistic regression by providing a *posterior* distribution of the noisy output measurement $y(x)$ at unobserved input $x \in \mathcal{X}$. The distribution of $y(x)$ at any input $x \in \mathcal{X}$ is a Gaussian distribution with the following posterior mean and

variance [Rasmussen and Williams, 2006]:

$$\begin{aligned}\mu_{t+1}(x) &\triangleq K_{x\mathbf{x}_{1:t}}(K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}}+\sigma_n^2 I)^{-1}\mathbf{y}_{1:t}\\ \sigma^2_{t+1}(x) &\triangleq k_{xx}+\sigma_n^2-K_{x\mathbf{x}_{1:t}}(K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}}+\sigma_n^2 I)^{-1}K_{\mathbf{x}_{1:t}x},\end{aligned} \tag{3.2}$$

in which $K_{x\mathbf{x}_{1:t}} \triangleq (k_{xx'})_{x'\in\mathbf{x}_{1:t}}$ is a row vector, vector $K_{\mathbf{x}_{1:t}x} \triangleq K_{x\mathbf{x}_{1:t}}^\top$ is the transpose of $K_{x\mathbf{x}_{1:t}}$, and matrix $K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} \triangleq (k_{x'x''})_{x',x''\in\mathbf{x}_{1:t}}$.

A key property of the GP model is that, different from μ_{t+1}, σ^2_{t+1} is independent of the output measurements $\mathbf{y}_{1:t}$.

3.2 GP-UCB algorithm

GP-UCB algorithm [Srinivas *et al.*, 2010] has become a classic BO algorithm due to its simplicity and strong theoretical performance guarantees. Its popularity resulted in a great number of works providing extensions and generalizations [Krause and Ong, 2011; Contal *et al.*, 2013; Desautels *et al.*, 2014; Kusner *et al.*, 2015; Ling *et al.*, 2016; Bogunovic *et al.*, 2016; Daxberger and Low, 2017; Dai *et al.*, 2019; Sessa *et al.*, 2019]. Since our works in Chapter 3 and Chapter 4 draw inspiration from GP-UCB algorithm as well, we review it in the following section of the thesis.

The AF adopted by the GP-UCB algorithm [Srinivas *et al.*, 2010] is the *upper confidence bound* (UCB) of the objective function f induced by the posterior GP model. In each iteration t, an input $x_t \in \mathcal{X}$ is selected to query by trading off between *(a)* sampling close to an expected maximum (i.e., with large posterior mean $\mu_t(x_t)$) given the current GP belief (i.e., exploitation) vs. *(b)* sampling an input with high predictive uncertainty (i.e., with large posterior standard deviation $\sigma_t(x_t)$) to improve the GP belief of f over $\mathcal{X}$ (i.e., exploration). Specifically,

$$x_t \triangleq \operatorname{argmax}_{x\in\mathcal{X}}\ \mu_t(x)+\beta_t^{1/2}\sigma_t(x),$$

in which the parameter $\beta_t > 0$ is set to trade off between exploitation vs. exploration. A remarkable property of the GP-UCB algorithm shown by the work of Srinivas *et al.* [2010] is that it achieves *no regret* asymptotically if the parameters $\beta_t > 0$ are chosen properly:

Theorem 3.1 (Srinivas *et al.* [2010])**.** *Let $\delta_{ucb} \in (0,1)$ and $\beta_t = 2\log(|\mathcal{X}|t^2\pi^2/6\delta_{ucb})$. Running GP-UCB with β_t for a sample f of a GP with mean function zero and covariance function $k(x,x')$, we obtain a regret bound of $\mathcal{O}(\sqrt{T\gamma_T \log(|\mathcal{X}|)})$ with high probability. Precisely,*

$$P(R_T \leq \sqrt{C_1 T \beta_T \gamma_T} \quad \forall T \geq 1) \geq 1 - \delta_{ucb}$$

where $C_1 \triangleq 8/\log(1+\sigma_n^{-2})$, $\gamma_T \triangleq \max_{\mathbf{x}_{1:T}\subset\mathcal{X}} \mathbb{I}[f(\mathcal{X}); \mathbf{y}_{1:T}]$ and $f(\mathcal{X}) \triangleq \{f(x)\}_{x\in\mathcal{X}}$.

Srinivas *et al.* [2010] also provide a bound for the maximum information gain γ_T on the function f from any set of noisy output measurements of size T, which, together with Theorem 3.1 above, result in the asymptotic no-regret property of GP-UCB algorithm:

Theorem 3.2 (Srinivas *et al.* [2010])**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex, $d \in \mathbb{N}$. Assume the covariance function satisfies $k(x,x') \leq 1$. Then*

1. *For the linear kernel $\gamma_T = \mathcal{O}(d \log T)$.*
2. *For the non-isotropic SE kernel $\gamma_T = \mathcal{O}\big((\log T)^{d+1}\big)$.*
3. *For the Matérn kernel with $\nu > 1$ $\gamma_T = \mathcal{O}(T^{\frac{d(d+1)}{2\nu+d(d+1)}} \log T)$.*

3.3 Problem setting

Privacy-preserving BO in the outsourced setting involves two parties: the *curator* who holds the sensitive dataset (e.g., a list of medical records), and the *modeler* who

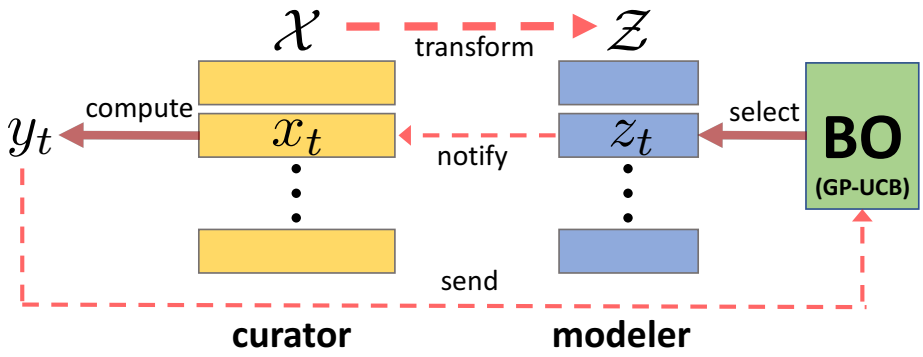


Figure 3.1: Visual illustration of the problem setting of outsourced BO.

performs the outsourced BO on the transformed dataset provided by the curator (see Fig. 3.1 for a visual illustration of this setting). The curator holds the original dataset represented as a set $\mathcal{X} \subset \mathbb{R}^d$ formed by n d-dimensional inputs. The curator and the modeler intend to maximize an unknown expensive-to-evaluate objective function f defined over $\mathcal{X}$. At the beginning, the curator performs a privacy-preserving transformation of the original dataset $\mathcal{X}$ to obtain a transformed dataset $\mathcal{Z} \subset \mathbb{R}^r$ formed by n r-dimensional inputs. As a result, every original input $x \in \mathcal{X}$ has an image, which is the corresponding transformed input $z \in \mathcal{Z}$. Then, the curator releases the transformed dataset $\mathcal{Z}$ to the modeler, who can subsequently start to run the BO algorithm on $\mathcal{Z}$. We assume that the BO procedure uses a GP with an isotropic[1] SE covariance function $k_{xx'}$ and zero mean μ_x. Furthermore, without loss of generality, we assume $k_{xx'} \leq 1$ for all $x, x' \in \mathcal{X}$.

In each iteration $t = 1, \ldots, T$, the modeler selects a transformed input $z_t \in \mathcal{Z}$ to query and notifies the curator about the choice of z_t. Next, the curator identifies x_t which is the preimage of z_t under the privacy-preserving transformation[2], and then computes $f(x_t)$ to yield a noisy output measurement: $y_t \triangleq f(x_t) + \epsilon_{GP}$, in which

[1] As pointed out in Section 3.1.2, non-isotropic covariance functions can be easily transformed to isotropic ones.

[2] We assume that $\mathcal{X}$ and $\mathcal{Z}$ describe the entire optimization domain, i.e., every $z_t \in \mathcal{Z}$ has a preimage $x_t \in \mathcal{X}$.

$\epsilon_{GP} \sim \mathcal{N}(0, \sigma_n^2)$ is a zero-mean Gaussian noise with noise variance σ_n^2. We assume that y_t is unknown to the curator in advance and is computed only when requested by the modeler, which is reasonable in all motivating scenarios in Section 1.1. The curator then sends y_t to the modeler for performing the next iteration of BO. We have assumed that in contrast to the input x_t, the noisy output measurement y_t does not contain sensitive information and can thus be non-privately released. This assumption is reasonable in our setting, e.g., if y_t represents the outcome of a medical test, revealing y_t does not unveil the identity of the patient. We leave the extension of privately releasing y_t for future work and briefly discuss it in Section 6.2.1.

3.4 Differential privacy

Differential privacy [Dwork *et al.*, 2006] has become the state-of-the-art technique for private data release. DP is a cryptographic framework which provides rigorous mathematical guarantees on privacy, typically by adding some random noise during the execution of the data release algorithm. DP has been widely adopted by the ML community for such methods as support vector machines [Rubinstein *et al.*, 2012], decision trees [Jagannathan *et al.*, 2012], Gaussian Processes [Smith *et al.*, 2018] and deep neural networks [Abadi *et al.*, 2016], among others. See the work of Sarwate and Chaudhuri [2013] for a detailed survey on applications of DP in ML.

Intuitively, DP promises that changing a single input of the dataset imposes only a small change in the output of the data release algorithm, hence the output does not depend significantly on any individual input. As a result, an attacker is not able to tell if an input is changed in the dataset just by looking at the output of the data release algorithm.

Randomization is essential for achieving DP: all DP algorithms include randomness. For completeness, we include the definition of a randomized algorithm and a

probability simplex necessary to define the former [Dwork and Roth, 2014]:

Definition 3.2. *Given a discrete set B, the probability simplex over B, denoted $\Delta(B)$, is defined to be:*

$$\Delta(B) \triangleq \left\{ x \triangleq (x_1, \dots x_{|B|}) \in \mathbb{R}^{|B|} : \ x_i \geq 0 \, \textit{for all} \, x_i \ \textit{and} \ \sum_{i=1}^{|B|} x_i = 1 \right\}.$$

Definition 3.3. *A randomized algorithm $\mathcal{M}$ with domain A and discrete range B is associated with a mapping $M : A \to \Delta(B)$. On input $a \in A$, the algorithm $\mathcal{M}$ outputs $\mathcal{M}(a) = b$ with probability $(M(a))_b$ for each $b \in B$. The probability space is over the coin flips of the algorithm $\mathcal{M}$.*

To define DP, we also need to introduce the notion of *neighboring* datasets. Following the prior works on DP [Blocki *et al.*, 2012; Hardt and Roth, 2012], we define two neighboring datasets as those differing only in a single row (i.e., a single input) with the norm of the difference bounded by 1:

Definition 3.4. *Let $\mathcal{X}, \mathcal{X}' \in \mathbb{R}^{n \times d}$ denote two datasets viewed as matrices[3] with d-dimensional inputs $\{x_{(i)}\}_{i=1}^n$ and $\{x'_{(i)}\}_{i=1}^n$ as rows respectively. We call datasets $\mathcal{X}$ and $\mathcal{X}'$ neighboring if there exists an index $i^* \in 1, \dots, n$ such that $\|x_{(i^*)} - x'_{(i^*)}\| \leq 1$, and $\|x_{(j)} - x'_{(j)}\| = 0$ for any index $j \in 1, \dots, n$, $j \neq i^*$.*

A randomized algorithm is differentially private if, for any two neighboring datasets, the distributions of the outputs of the algorithm calculated on these datasets are similar. Formally:

Definition 3.5. *A randomized algorithm $\mathcal{M}$ is (ϵ, δ)-differentially private for $\epsilon > 0$ and $\delta \in (0, 1)$ if, for all $O \subset Range(\mathcal{M})$ (where $Range(\mathcal{M})$ is the range of the outputs*

[3] We slightly abuse the notation and view the dataset $\mathcal{X}$ ($\mathcal{Z}$) as an $n \times d$ ($n \times r$) matrix where each of the n rows corresponds to an original (transformed) input.

of the randomized algorithm $\mathcal{M}$) and for all neighboring datasets $\mathcal{X}$ and $\mathcal{X}'$, we have

$$P(\mathcal{M}(\mathcal{X}) \in O) \leq \exp(\epsilon) \cdot P(\mathcal{M}(\mathcal{X}') \in O) + \delta.$$

Note that the definition above is symmetric in terms of $\mathcal{X}$ and $\mathcal{X}'$. A $(\epsilon, 0)$-differentially private algorithm is usually called ϵ-differentially private. The DP parameters ϵ, δ control the *privacy-utility trade-off*: The smaller they are, the tighter the privacy guarantee is, at the expense of lower accuracy due to the increased amount of noise required to satisfy DP. The DP parameter δ is usually set smaller than $1/n$ where n is the number of inputs in the dataset [Dwork and Roth, 2014; Abadi *et al.*, 2016; Foulds *et al.*, 2016; Papernot *et al.*, 2017a], while the DP parameter ϵ is usually set in the single-digit range, as can be seen from the state-of-the-art works on the application of DP in machine learning [Abadi *et al.*, 2016; Foulds *et al.*, 2016; Papernot *et al.*, 2017a].

3.4.1 Common DP mechanisms

In this section we review some common DP techniques, such as Laplace and Gaussian mechanisms. For these mechanisms we first define an intermediate quantity called the *sensitivity* describing how much the output of the objective function f changes on the neighboring datasets:

Definition 3.6. *The ℓ_p-sensitivity of a function f is defined as*

$$\Delta_p f \triangleq \max_{\mathcal{X}, \mathcal{X}'} \|f(\mathcal{X}) - f(\mathcal{X}')\|_p$$

where $\mathcal{X}$ and $\mathcal{X}'$ are neighboring datasets and $\|\cdot\|_p$ is ℓ_p norm.

The sensitivity of a function f captures the magnitude by which a single individual's data can change the output of function f in the worst case. Therefore, this

quantity is related to the amount of noise required to hide such a change in the dataset from the attacker who observes the output of the DP mechanism. We now show how sensitivity can be used to construct DP mechanisms for releasing the output of function f.

Laplace mechanism. This mechanism computes f and perturbs each coordinate with noise drawn from the Laplace distribution with the scale proportional to the l_1-sensitivity of function f:

Definition 3.7. *Given any function f computed over a dataset $\mathcal{X}$ with a value in $\mathbb{R}^k$ the Laplace mechanism is defined as:*

$$\mathcal{M}_L(\mathcal{X}, f, \epsilon) \triangleq f(x) + (Y_1, \ldots, Y_k)$$

where Y_i are i.i.d. random variables drawn from Laplace distribution Laplace$(\Delta_1 f/\epsilon)$.

Theorem 3.3. *The Laplace mechanism preserves ϵ-differential privacy.*

See Section 3.3 of [Dwork and Roth, 2014] for the proof.

Gaussian mechanism. This mechanism is similar to the Laplace mechanism, but uses Gaussian noise instead of Laplace noise. Additionally, the noise added is proportional to the l_2-sensitivity of f and not to the l_1-sensitivity, as in the previous case:

Definition 3.8. *Given any function f computed over a dataset $\mathcal{X}$ with a value in $\mathbb{R}^k$ the Gaussian mechanism is defined as:*

$$\mathcal{M}_G(\mathcal{X}, f, \sigma) \triangleq f(x) + (Y_1, \ldots, Y_k)$$

where Y_i are i.i.d. random variables drawn from Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

Theorem 3.4. *Let $\epsilon > 0$ and $\delta \in (0, 1)$ be given. For $c^2 > \ln(1.25/\delta)$ the Gaussian mechanism with parameter $\sigma \geq c\Delta_2 f/\epsilon$ is (ϵ, δ)-differentially private.*

See Appendix A of [Dwork and Roth, 2014] for the proof.

Note that despite the similar formulation, Gaussian and Laplace mechanisms have certain distinctions: They use different sensitivities (l_1 vs. l_2) and satisfy different DP guarantees (ϵ-DP vs. (ϵ, δ)-DP). Refer to the work of [Dwork and Roth, 2014] for more details about DP.

3.5 Private Outsourced Bayesian Optimization

In our PO-GP-UCB algorithm, the curator needs to perform a privacy-preserving transformation of the original dataset $\mathcal{X} \subset \mathbb{R}^d$ and release the transformed dataset $\mathcal{Z} \subset \mathbb{R}^r$ to the modeler. Subsequently, the modeler runs BO (i.e., GP-UCB) using $\mathcal{Z}$. When performing the transformation, the goal of the curator is two-fold: Firstly, the transformation has to be differentially private with given DP parameters ϵ, δ (Definition 3.5); secondly, the transformation should allow the modeler to obtain good BO performance on the transformed dataset (in a sense to be formalized later in this section).

3.5.1 Transformation via Random Projection

Good BO performance by the modeler (i.e., the second goal of the curator) can be achieved by making the GP predictions (3.2) (on which the performance of the BO algorithm depends) using the transformed dataset $\mathcal{Z}$ close to those using the original dataset $\mathcal{X}$. To this end, we ensure that the distances between all pairs of inputs are approximately preserved after the transformation. This is motivated by the fact that the GP predictions (3.2) and hence the BO performance, depend on the inputs only through the value of covariance, which, in the case of isotropic covariance functions[1], only depends on the pairwise distances between inputs. Consequently, by

preserving the pairwise distances between inputs, the performance of the BO (GP-UCB) algorithm run by the modeler on $\mathcal{Z}$ is made similar to that of the non-private GP-UCB algorithm run on the original dataset $\mathcal{X}$, for which theoretical convergence guarantee has been shown [Srinivas *et al.*, 2010]. As a result, the BO performance in the outsourced setting can be theoretically guaranteed (Section 3.5.3) and thus practically assured.

Therefore, to achieve both goals of the curator, we need to address the question as to *what transformation preserves both the pairwise distances between inputs and DP*. A natural approach is to add noise directly to the matrix of pairwise distances between the original inputs from $\mathcal{X}$ using standard DP methods such as the Laplace or Gaussian mechanisms (Section 3.4.1). However, the resulting noisy distance matrix is not guaranteed to produce an invertible covariance matrix $K_{\mathcal{X}_t\mathcal{X}_t} + \sigma_n^2 I$, which is a requirement for the GP predictions (3.2). Instead, we perform the transformation through a technique based on random projection, which satisfies both goals of the curator. Firstly, random projection through random samples from standard normal distribution has been shown to preserve DP [Blocki *et al.*, 2012]. Secondly, as a result of the Johnson-Lindenstrauss lemma [Johnson and Lindenstrauss, 1984], random projection is also able to approximately preserve the pairwise distances between inputs, as shown in the following lemma:

Lemma 3.1. *Let $\nu \in (0, 1/2)$, $\mu \in (0, 1)$, $d \in \mathbb{N}$ and a set $\mathcal{X} \subset \mathbb{R}^d$ of n row vectors be given. Let $r \in \mathbb{N}$ and M be a $d \times r$ matrix whose entries are i.i.d. samples from $\mathcal{N}(0, 1)$. If $r \geq 8 \log(n^2/\mu)/\nu^2$, the probability of*

$$(1 - \nu)\|x - x'\|^2 \leq r^{-1}\|xM - x'M\|^2 \leq (1 + \nu)\|x - x'\|^2$$

for all $x, x' \in \mathcal{X}$ is at least $1 - \mu$.

Remark 3.1. Parameter r controls the dimension of the random projection, while parameters ν and μ control the accuracy. Lemma 3.1 corroborates the intuition that a smaller value of r leads to larger values of ν and μ, i.e., lower random projection accuracy.

The proof (Appendix A.1) consists of a union bound applied to the Johnson-Lindenstrauss lemma [Johnson and Lindenstrauss, 1984], which is a result from geometry stating that a set of points in a high-dimensional space can be embedded into a lower-dimensional space such that the pairwise distances between the points are nearly preserved. The lemma has been used in many domains of computer science such as graph embeddings [Linial *et al.*, 1995], information retrieval [Papadimitriou *et al.*, 1995], compressed sensing [Baraniuk *et al.*, 2008] and ML [Balcan *et al.*, 2006]. Formally, it is stated below.

Theorem 3.5. *[Johnson-Lindenstrauss lemma [Johnson and Lindenstrauss, 1984]] Let $\nu \in (0, 1/2)$, $r \in \mathbb{N}$ and $d \in \mathbb{N}$ be given. Let M' be a $r \times d$ matrix whose entries are i.i.d. samples from $\mathcal{N}(0, 1)$. Then for any vector $y \in \mathbb{R}^d$*

$$P\Big((1-\nu)\|y\|^2 \leq r^{-1}\|M'y\|^2 \leq (1+\nu)\|y\|^2\Big) \geq 1 - 2\exp(-\nu^2 r/8).$$

We now design a DP dataset transformation based on the random projection.

3.5.2 The Curator Part

The curator part (Algorithm 2) of our PO-GP-UCB algorithm takes as input the original dataset $\mathcal{X}$ viewed as an $n \times d$ matrix[3], the DP parameters ϵ, δ (Definition 3.5) and the random projection parameter r (Lemma 3.1)[4]. To begin with, the curator

[4] Note that in Theorem 3.8, the parameter r is calculated based on specific values of the parameters μ and ν (Lemma 3.1) in order to achieve the performance guarantee. However, in practice, μ and ν are not required to specify the value of r for Algorithm 2.

subtracts the mean from each column of $\mathcal{X}$ (line 2), and then picks a matrix M of samples from standard normal distribution $\mathcal{N}(0,1)$ to perform random projection (line 3). Next, if the smallest singular value $\sigma_{min}(\mathcal{X})$ of the centered dataset $\mathcal{X}$ is not less than a threshold ω (calculated in line 5), the curator outputs the random projection $\mathcal{Z} \triangleq r^{-1/2}\mathcal{X}M$ of the centered dataset $\mathcal{X}$ (line 7). Otherwise, the curator increases the singular values of the centered dataset $\mathcal{X}$ (line 9) to obtain a new dataset $\tilde{\mathcal{X}}$ and outputs the random projection $\mathcal{Z} \triangleq r^{-1/2}\tilde{\mathcal{X}}M$ of the new dataset $\tilde{\mathcal{X}}$ (line 10). Lastly, the curator releases $\mathcal{Z}$ to the modeler (line 11).

Algorithm 2 PO-GP-UCB (The curator part)

1: **Input:** $\mathcal{X}$, ϵ, δ, r
2: $\mathcal{X} \leftarrow \mathcal{X} - \mathbf{1}\mathbf{1}^\top\mathcal{X}/n$ where $\mathbf{1}$ is a $n \times 1$ vector of 1's
3: Pick a $d \times r$ matrix M of i.i.d. samples from $\mathcal{N}(0,1)$
4: Compute the SVD of $\mathcal{X} = U\Sigma V^\top$
5: $\omega \leftarrow 16\sqrt{r\log(2/\delta)}\epsilon^{-1}\log(16r/\delta)$
6: **if** $\sigma_{min}(\mathcal{X}) \geq \omega$ **then**
7: **return** $\mathcal{Z} \leftarrow r^{-1/2}\mathcal{X}M$
8: **else**
9: $\tilde{\mathcal{X}} \leftarrow U\sqrt{\Sigma^2 + \omega^2 I_{n\times d}}V^\top$ where Σ^2 $(I_{n\times d})$ is an $n \times d$ matrix whose main diagonal has squared singular values of $\mathcal{X}$ (ones) in each coordinate and all other coordinates are 0
10: **return** $\mathcal{Z} \leftarrow r^{-1/2}\tilde{\mathcal{X}}M$
11: Release dataset $\mathcal{Z}$ to the modeler

The fact that Algorithm 2 both preserves DP and approximately preserves the pairwise distances between inputs is stated in Theorems 3.6 and 3.7 below.

Theorem 3.6. *Algorithm 2 preserves (ϵ, δ)-DP.*

In the proof of Theorem 3.6 (Appendix A.2), all singular values of the dataset $\mathcal{X}$ are required to be not less than ω (calculated in line 5). This explains the necessity of line 9, where we increase the singular values of the dataset $\mathcal{X}$ if $\sigma_{min}(\mathcal{X}) < \omega$, to ensure that this requirement is satisfied.

Theorem 3.7. *Let a dataset $\mathcal{X} \subset \mathbb{R}^d$ be given. Let $\nu \in (0, 1/2)$, $\mu \in (0, 1)$ be given. Let $r \in \mathbb{N}$, such that $r \geq 8 \log(n^2/\mu)/\nu^2$. Then, the probability of*

$$(1-\nu)\|x - x'\|^2 \leq \|z - z'\|^2 \leq (1+\nu)C'\|x - x'\|^2$$

for all $x, x' \in \mathcal{X}$ and their images $z, z' \in \mathcal{Z}$ is at least $1 - \mu$, in which $C' \triangleq 1 + \mathbb{1}_{\sigma_{min}(\mathcal{X}) < \omega}\omega^2/\sigma^2_{min}(\mathcal{X})$.

The proof (see Appendix A.3) consists of bounding the change in distances between inputs due to the increase of the singular values of the dataset $\mathcal{X}$ (line 9 of Algorithm 2) and applying Lemma 3.1. It can be observed from Theorem 3.7 that when $\sigma_{min}(\mathcal{X}) \geq \omega$, $C' = 1$ and hence Algorithm 2 approximately preserves the pairwise distances between inputs.

There are several important differences between our Algorithm 2 and the work of Blocki *et al.* [2012]. Firstly, Algorithm 3 of Blocki *et al.* [2012] releases a DP estimate of the dataset covariance matrix, while our Algorithm 2 outputs a DP transformation of the original dataset. Secondly, Algorithm 3 of Blocki *et al.* [2012] does not have the "if/else" condition (line 6 of Algorithm 2) and always increases the singular values as in line 9 of Algorithm 2. In our case, however, if the singular values are increased due to the condition $\sigma_{min}(\mathcal{X}) < \omega$ (i.e., the "else" clause, line 8 of Algorithm 2), the pairwise input distances of the dataset $\mathcal{X}$ are no longer approximately preserved in $\mathcal{Z}$ (Theorem 3.7), which results in a slightly different regret bound (see Theorem 3.8 and Remark 3.2 below). This requires us to introduce the "if/else" condition in Algorithm 2. We discuss these changes in greater detail in Appendix A.2.

3.5.3 The Modeler Part

The modeler part of our PO-GP-UCB algorithm (Algorithm 3) takes as input the transformed dataset $\mathcal{Z} \subset \mathbb{R}^r$ received from the curator as well as the GP-UCB parameter δ', and runs the GP-UCB algorithm for T iterations on $\mathcal{Z}$. In each iteration t, the modeler selects the candidate transformed input z_t by maximizing the GP-UCB AF (line 4), and queries the curator for the corresponding noisy output measurement y_t (line 5). To perform such a query, the modeler can send the index (row) i_t of the selected transformed input z_t in the dataset $\mathcal{Z}$ viewed as a matrix[3] to the curator. The curator can then find the preimage x_t of z_t by looking into the same row i_t of the dataset $\mathcal{X}$ viewed as a matrix[3]. After identifying x_t, the curator can compute $f(x_t)$ to yield a noisy output measurement $y_t \triangleq f(x_t) + \epsilon_{GP}$ and send it to the modeler. The modeler then updates the GP posterior belief (line 6) and proceeds to the next iteration $t+1$.

Algorithm 3 PO-GP-UCB (The modeler part)

1: **Input:** $\mathcal{Z}$, δ', T
2: **for** $t = 1, \ldots, T$ **do**
3: Set $\beta_t \leftarrow 2\log(nt^2\pi^2/6\delta')$
4: $z_t \leftarrow \text{argmax}_{z\in\mathcal{Z}}\, \tilde{\mu}_t(z) + \beta_t^{1/2}\tilde{\sigma}_t(z)$
5: Query the curator for y_t
6: Update GP posterior belief: $\tilde{\mu}_{t+1}(z)$ and $\tilde{\sigma}_{t+1}(z)$

In our theoretical analysis, we make the assumption of the *diagonal dominance* property of the covariance matrices, which was also used by previous works on GP with DP [Smith *et al.*, 2018] and active learning [Hoang *et al.*, 2014]:

Definition 3.9. *Let a dataset $\mathcal{X} \subset \mathbb{R}^d$ and a set $\mathcal{X}_0 \subseteq \mathcal{X}$ be given. The covariance matrix $K_{\mathcal{X}_0\mathcal{X}_0}$ is said to be diagonally dominant if for any $x \in \mathcal{X}_0$*

$$k_{xx} \geq \left(\sqrt{|\mathcal{X}_0| - 1} + 1\right) \sum\nolimits_{x' \in \mathcal{X}_0 \setminus x} k_{xx'}.$$

Note that this assumption is adopted mainly for the theoretical analysis, and is thus not strictly required in order for our algorithm to deliver competitive practical performance (Section 3.6). Theorem 3.8 below presents the theoretical guarantee on the BO performance of our PO-GP-UCB algorithm run by the modeler (Algorithm 3).

Theorem 3.8. *Let* $\varepsilon_{ucb} > 0$, $\delta_{ucb} \in (0,1)$, $T \in \mathbb{N}$, *DP parameters* ϵ *and* δ, *and a dataset* $\mathcal{X} \subset \mathbb{R}^d$ *be given. Let* $\mathrm{d} \triangleq diam(\mathcal{X})/l$ *where* $diam(\mathcal{X})$ *is the diameter of* $\mathcal{X}$ *and* l *is the GP length-scale. Suppose for all* $t = 1, \ldots, T$, $|y_t| \leq L$ *and* $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$ *is diagonally dominant. Suppose* $r \geq 8\log(n^2/\mu)/\nu^2$ *(Algorithm 2) where* $\mu \triangleq \delta_{ucb}/2$ *and* $\nu \triangleq \min(\varepsilon_{ucb}/(2\sqrt{3}\mathrm{d}^2 L), 2/\mathrm{d}^2, 1/2)$, *and* $\delta' \triangleq \delta_{ucb}/2$ *(Algorithm 3). If* $\sigma_{min}(\mathcal{X}) \geq \omega$, *then the simple regret* S_T *incurred by Algorithm 3 run by the modeler satisfies*

$$S_T \leq \left(\varepsilon_{ucb}^2 + 24(C_2 + C_1\beta_T^{1/2})^2 \log T/T + 24/\log(1+\sigma_n^{-2}) \cdot \beta_T\gamma_T/T\right)^{1/2}$$

with probability at least $1 - \delta_{ucb}$, *in which* γ_T *is the maximum information gain on the function* f *from any set of noisy output measurements of size* T, $C_1 \triangleq \mathcal{O}\big(\sigma_y\sqrt{\sigma_y^2+\sigma_n^2}(\sigma_y^2/\sigma_n^2+1)\big)$ *and* $C_2 \triangleq \mathcal{O}(\sigma_y^2/\sigma_n^2 \cdot L)$.

The key idea of the proof (Appendix A.5) is to ensure that every value of the GP-UCB AF computed on the transformed dataset $\mathcal{Z}$ is close to the value of the corresponding GP-UCB AF computed on the original dataset $\mathcal{X}$. Consequently, the regret of the PO-GP-UCB algorithm run on $\mathcal{Z}$ can be analyzed using similar techniques as those adopted in the analysis of the non-private GP-UCB algorithm run on the original dataset $\mathcal{X}$ [Srinivas *et al.*, 2010], which leads to the regret bound shown in Theorem 3.8. Note that Srinivas *et al.* [2010] has shown that $\gamma_T = \mathcal{O}((\log T)^{d+1})$ for the squared exponential kernel (see Theorem 3.2 in Section 3.2).

Remark 3.2. If $\sigma_{min}(\mathcal{X}) < \omega$, a similar upper bound on the regret can be proved with the difference that ε_{ucb} specified by the curator is replaced by a different constant, which, unlike ε_{ucb}, cannot be set arbitrarily. This results from the fact that

if $\sigma_{min}(\mathcal{X}) < \omega$, Algorithm 2 increases the singular values of the dataset $\mathcal{X}$ (see line 9). As a consequence, the pairwise distances between inputs are no longer approximately preserved after the transformation (see Theorem 3.7), resulting in a looser regret bound (see Remark A.2 in Appendix A.5).

Remark 3.3. The presence of the constant ε_{ucb} makes the regret upper bound of PO-GP-UCB slightly different from that of the original GP-UCB algorithm. ε_{ucb} can be viewed as controlling the trade-off between utility (BO performance) and privacy preservation (see more detailed discussion in Section 3.5.4). In contrast, the only prior works on privacy-preserving BO by Kusner *et al.* [2015] and Nguyen *et al.* [2018] do not provide any regret bounds.

Remark 3.4. The upper bound on the simple regret S_T in Theorem 3.8 indirectly depends on the DP parameter ϵ: the bound holds under the condition $\sigma_{min}(\mathcal{X}) \geq \omega$, in which ω depends on ϵ (line 5 of Algorithm 2). Moreover, when $\sigma_{min}(\mathcal{X}) < \omega$, ε_{ucb} (which appears in the regret bound) is replaced by a different constant, which depends on ϵ (see Remark 3.2).

3.5.4 Analysis and Discussion

Interestingly, our theoretical results are amenable to elegant interpretations regarding the privacy-utility trade-off.

The flexibility to tune the value of ω to satisfy the condition required by Theorem 3.8 (i.e., $\sigma_{min}(\mathcal{X}) \geq \omega$) incurs an interesting trade-off. Specifically, if $\sigma_{min}(\mathcal{X}) < \omega$, we have two choices:

(a) to run PO-GP-UCB without modifying any parameter;

(b) to reduce ω by tuning the algorithmic parameters to satisfy the condition $\sigma_{min}(\mathcal{X}) \geq \omega$.

Both of these choices incur some costs. In case (a), the resulting regret bound is looser as explained in Remark 3.2, which might imply worse BO performance. In case (b), to reduce the value of ω, we can again have two options:

(i) to increase the DP parameters ϵ and δ which deteriorates the DP guarantee;

(ii) decrease the value of r. A smaller value of r implies larger values of μ and ν as required by Theorem 3.8 ($r \geq 8\log(n^2/\mu)/\nu^2$) and thus larger values of ε_{ucb} and δ_{ucb} as seen in the definitions of μ and ν in Theorem 3.8.

This consequently results in a worse regret upper bound (Theorem 3.8) and thus deteriorated BO performance. Therefore, the privacy-utility trade-off is involved in our strategy to deal with the scenario where $\sigma_{min}(\mathcal{X}) < \omega$.

For a fixed value of ω such that $\sigma_{min}(\mathcal{X}) \geq \omega$, the privacy-utility trade-off can also be identified and thus utilized to adjust the the algorithmic parameters: $\epsilon, \delta, \varepsilon_{ucb}$ and δ_{ucb}. Specifically, decreasing the values of the DP parameters ϵ and δ improves the privacy guarantee. However, in order to fix the value of ω (to ensure that the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ remains satisfied), the value of r needs to be reduced, which results in larger values of ε_{ucb} and δ_{ucb} and thus worse BO performance (as discussed in the previous paragraph). Similar analysis reveals that decreasing the values of ε_{ucb} and δ_{ucb} improves the BO performance, at the expense of looser privacy guarantee (i.e., larger required values of ϵ and δ). Furthermore, the role played by ω in Algorithm 2 provides a guideline on the practical design of the algorithm. In particular, for a fixed desirable level of privacy (i.e., fixed values of ϵ and δ), the value of r should be made as large as possible while still ensuring that the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is satisfied, since larger r improves the BO performance until this condition is violated. This guideline will be exploited and validated in the experiments.

These insights regarding the privacy-utility trade-off serve as intuitive justifications of our algorithm and provide useful guidelines for its practical deployment.

3.6 Experimental results

In this section, we empirically evaluate the performance of our PO-GP-UCB algorithm using four datasets:

- *Synthetic GP dataset.* The original inputs for this experiment are 2-dimensional vectors arranged into a uniform grid and discretized into a 100×100 input domain (i.e., $d = 2$ and $n = 10000$). The function to maximize is sampled from a GP with the GP hyperparameters $\mu_x = 0$, $l = 1.25$, $\sigma_y^2 = 1$ and $\sigma_n^2 = 10^{-5}$.

- *Real-world loan applications dataset.* A bank is selecting the loan applicants with the highest return on investment (ROI) and outsources the task to a financial AI consultancy. For this experiment we use the public data from https://www.lendingclub.com/. The inputs to BO are the data of 36000 loan applicants, each consisting of three features: the total amount committed by investors for the loan at that point in time, the interest rate on the loan and the annual income provided by the applicant during registration (i.e., $n = 36000$ and $d = 3$). The function to maximize (the output measurement) is the ROI for an applicant. The original ROI measurements are log-transformed to remove skewness and extremity for stabilizing the GP covariance structure and the GP hyperparameters $\mu_x = -2.742$, $l_1 = 18985.93$ dollars, $l_2 = 10.505$ percent, $l_3 = 171490.464$ dollars, $\sigma_y^2 = 2.118$ and $\sigma_n^2 = 0.83$ are then learned using maximum likelihood estimation [Rasmussen and Williams, 2006]. The original inputs are preprocessed to form an isotropic covariance function[1].

- *Real-world private property price dataset.* A real estate agency is trying to locate the cheapest private properties and outsources the task of selecting the candidate properties to an AI consultancy. The original inputs are the longitude/latitude coordinates of 2004 individual properties (i.e., $n = 2004$ and $d =$

2). We use the public data from https://www.ura.gov.sg/realEstateIIWeb/transaction/search.action. The function to minimize is the evaluated property price measured in dollars per square meter. The original property price measurements are log-transformed to remove skewness and extremity for stabilizing the GP covariance structure and the GP hyperparameters $\mu_x = 6.85$, $l = 0.555$, $\sigma_y^2 = 0.545$ and $\sigma_n^2 = 0.527$ are then learned using maximum likelihood estimation [Rasmussen and Williams, 2006].

- *Branin-Hoo benchmark function.* The original inputs for this experiment are 2-dimensional vectors arranged into a uniform grid and discretized into a 31×31 input domain (i.e., $d = 2$ and $n = 961$). The function to maximize is sampled from the negation of Branin-Hoo function. The original output measurements are log-transformed to remove skewness and extremity in order to stabilize the GP covariance structure. The GP hyperparameters are learned using maximum likelihood estimation [Rasmussen and Williams, 2006]. The original inputs are preprocessed to form an isotropic covariance function[1].

The performances of our algorithm are compared with that of the non-private GP-UCB algorithm run using the original datasets [Srinivas *et al.*, 2010]. The performance metric used is simple regret. All results are averaged over 50 random runs, each of which uses a different set of initializations for BO. Each random run uses an independent realization of the matrix M of i.i.d. samples from $\mathcal{N}(0, 1)$ for performing random projection (line 3 of Algorithm 2).

We set the GP-UCB parameter $\delta_{ucb} = 0.05$ (Theorem 3.8) and normalize the inputs to have a maximal norm of 25 in all experiments. Following the guidelines by the state-of-the-art works in DP [Dwork and Roth, 2014; Abadi *et al.*, 2016; Foulds *et al.*, 2016; Papernot *et al.*, 2017a], we fix the value of the DP parameter δ (Definition 3.5) to be smaller than $1/n$ in all experiments (see Section 3.4).

Note that setting the values of the parameters μ, ν (Lemma 3.1) and the GP-UCB parameter ε_{ucb} (Theorem 3.8), as well as assuming the diagonal dominance of covariance matrices (Definition 3.9) is required only for our theoretical analysis and thus not necessary in the practical employment of our algorithm.

In every experiment that varies the value of the DP parameter ϵ (Definition 3.5), the PO-GP-UCB algorithm with the largest value of ϵ under consideration satisfies the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ (i.e., the "if" clause, line 6 of Algorithm 2), while the algorithms with all other values of ϵ under consideration satisfy the condition $\sigma_{min}(\mathcal{X}) < \omega$ (i.e., the "else" clause, line 8 of Algorithm 2). This is explained by the fact that further increasing the value of ϵ will only decrease the value of ω (see line 5 of Algorithm 2), so the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ will remain satisfied. As a result, the dataset $\mathcal{Z}$ returned by Algorithm 2 and hence the performance of PO-GP-UCB will stay the same.

3.6.1 Synthetic GP dataset

For this experiment we set the parameter $r = 10$ (Algorithm 2), DP parameter $\delta = 10^{-5}$ (Definition 3.5) and the GP-UCB parameter $T = 50$.

Fig. 3.2 shows the performances of PO-GP-UCB with different values of ϵ and that of non-private GP-UCB. It can be observed that smaller values of ϵ (tighter privacy guarantees) result in larger simple regret, which is consistent with the privacy-utility trade off. PO-GP-UCB with the largest value of $\epsilon = \exp(1.1)$ satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ achieves only $0.011\sigma_y$ more simple regret than non-private GP-UCB after 50 iterations. Interestingly, despite having a looser regret bound (see Remark 3.2), the PO-GP-UCB algorithm with some smaller values of ϵ satisfying the condition $\sigma_{min}(\mathcal{X}) < \omega$ also only incurs slightly larger regret than non-private GP-UCB. In particular, PO-GP-UCB with $\epsilon = \exp(0.9)$ ($\epsilon = \exp(0.0)$) achieves only

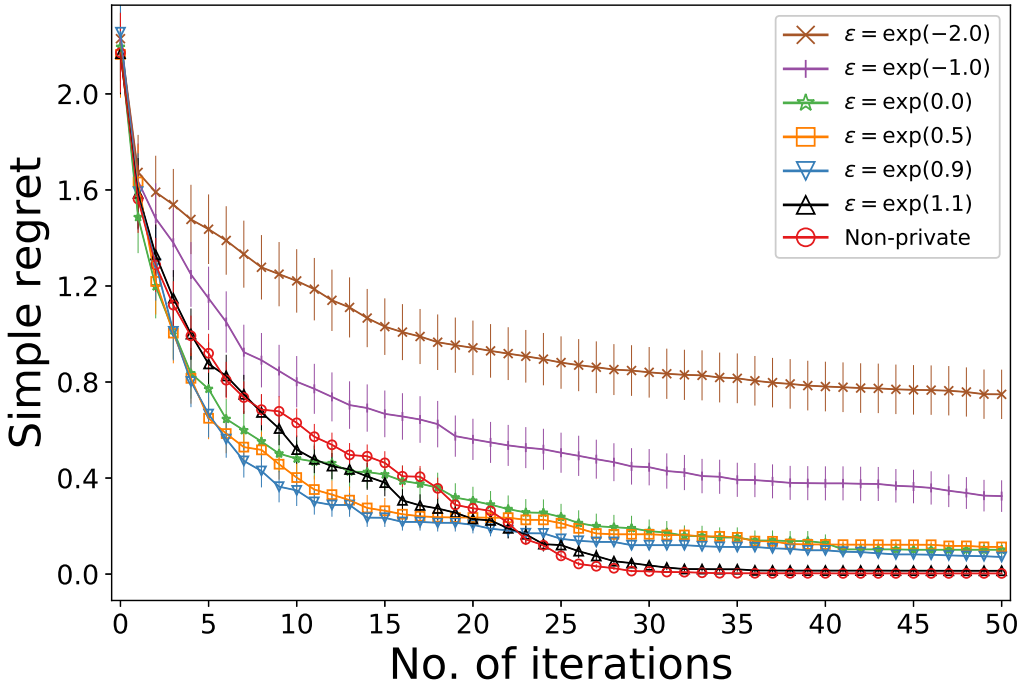


Figure 3.2: Simple regrets achieved by tested BO algorithms (with fixed r and different values of ϵ) vs. the number of iterations for the synthetic GP dataset, $r = 10$.

$0.069\sigma_y$ ($0.099\sigma_y$) more simple regret after 50 iterations. Therefore, our algorithm is able to achieve favorable performance with the values of ϵ in the single-digit range, which is consistent with the practice of the state-of-the-art works on the application of DP in machine learning [Abadi *et al.*, 2016; Foulds *et al.*, 2016; Papernot *et al.*, 2017a]. This implies our algorithm's practical capability of simultaneously achieving tight privacy guarantee and obtaining competitive BO performance.

We also investigate the impact of varying the value of the random projection parameter r on the performance of PO-GP-UCB. In particular, we consider 3 different values of DP parameter ϵ: $\epsilon = \exp(1.1)$, $\epsilon = \exp(1.3)$ and $\epsilon = \exp(1.5)$. We then fix the value of ϵ and vary the value of r. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 10$ for $\epsilon = \exp(1.1)$, $r = 15$ for $\epsilon = \exp(1.3)$ and

$r = 20$ for $\epsilon = \exp(1.5)$. Tables 3.1, 3.2 and 3.3 reveal that the largest values of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ lead to the smallest simple regret after 50 iterations. Decreasing the value of r increases the simple regret, which agrees with our analysis in Section 3.5.4 (i.e., smaller r results in worse regret upper bound). On the other hand, increasing r such that the condition $\sigma_{min}(\mathcal{X}) < \omega$ is satisfied also results in larger simple regret, which is again consistent with the analysis in Remark 3.2 stating that the regret upper bound becomes looser in this scenario. This experiment suggests that, in practice, for a fixed desirable privacy level (i.e., if the values of the DP parameters ϵ and δ are fixed), r should be chosen as the largest value satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$.

Table 3.1: Simple regrets achieved by PO-GP-UCB with fixed $\epsilon = \exp(1.1)$ and different values of r after 50 iterations for the synthetic GP dataset. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 10$.

r	3	6	8	10	15	20
S_{50}	0.073	0.038	0.018	**0.014**	0.118	0.137

Table 3.2: Simple regrets achieved by PO-GP-UCB with fixed $\epsilon = \exp(1.3)$ and different values of r after 50 iterations for the synthetic GP dataset. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 15$.

r	3	9	12	15	20	30
S_{50}	0.091	0.009	0.019	**0.008**	0.127	0.134

Table 3.3: Simple regrets achieved by PO-GP-UCB with fixed $\epsilon = \exp(1.5)$ and different values of r after 50 iterations for the synthetic GP dataset. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 20$.

r	5	10	15	20	30	50
S_{50}	0.05	0.021	0.003	**0.002**	0.094	0.142

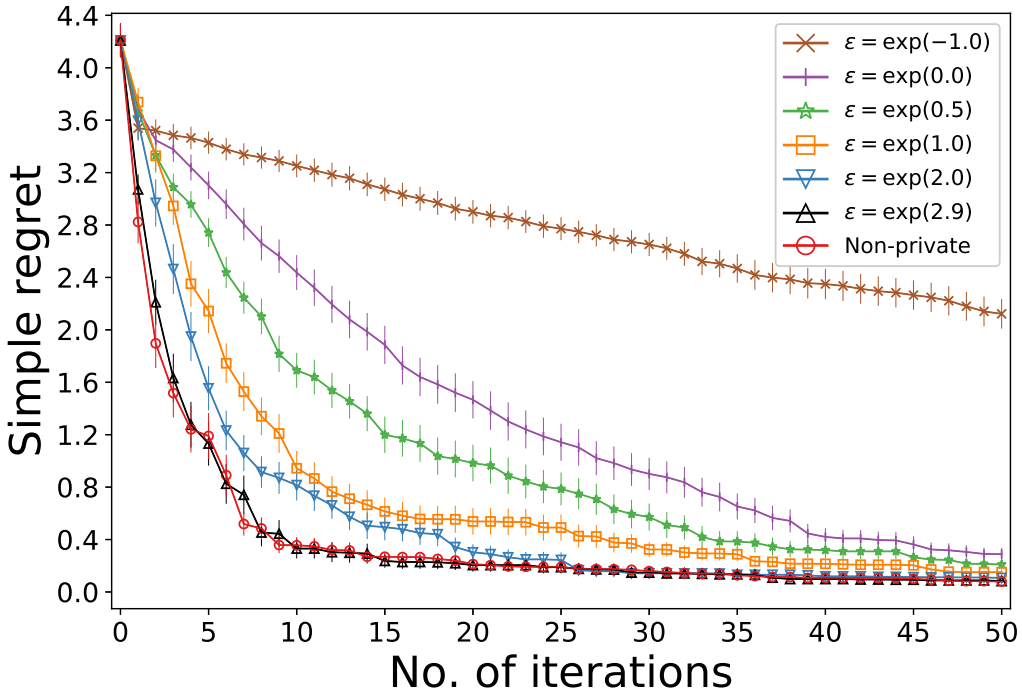


Figure 3.3: Simple regrets achieved by tested BO algorithms (with fixed r and different values of ϵ) vs. the number of iterations for loan applications dataset, $r = 15$.

3.6.2 Real-world loan applications dataset

For this experiment we set $r = 15$ (Algorithm 2), DP parameter $\delta = 10^{-5}$ (Definition 3.5) and the GP-UCB parameter $T = 50$.

Fig. 3.3 presents the results of varying the value of ϵ. Similar to the synthetic GP dataset, after 50 iterations, the simple regret achieved by PO-GP-UCB with the largest value of $\epsilon = \exp(2.9)$ satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is slightly larger (by $0.003\sigma_y$) than that achieved by non-private GP-UCB. Moreover, PO-GP-UCB with some values of ϵ in the single-digit range satisfying the condition $\sigma_{min}(\mathcal{X}) < \omega$ shows marginally worse performance compared with non-private GP-UCB. In particular, after 50 iterations, $\epsilon = \exp(2.0)$ and $\epsilon = \exp(1.0)$ result in $0.019\sigma_y$ and $0.05\sigma_y$ more simple regret than non-private GP-UCB respectively.

We examine the effect of r on the performance of PO-GP-UCB, by fixing the value of DP parameter ϵ and changing r. We consider 3 different values of DP parameter ϵ: $\epsilon = \exp(2.7)$, $\epsilon = \exp(2.9)$ and $\epsilon = \exp(3.1)$. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 10$ for $\epsilon = \exp(2.7)$, $r = 15$ for $\epsilon = \exp(2.9)$ and $r = 20$ for $\epsilon = \exp(3.1)$. The results are presented in Tables 3.4, 3.5 and 3.6. PO-GP-UCB with the largest r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ in general leads to the best performance, i.e., it achieves the smallest simple regret in Tables 3.4 and 3.5, and the second smallest simple regret in Table 3.6. Similar insights to the results of the synthetic GP dataset can also be drawn: reducing the value of r and increasing the value of r to satisfy the condition $\sigma_{min}(\mathcal{X}) < \omega$ both result in larger simple regret, which again corroborates our theoretical analysis.

Table 3.4: Simple regrets achieved by PO-GP-UCB with fixed $\epsilon = \exp(2.7)$ and different values of r after 50 iterations for the real-world loan applications dataset. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 10$.

r	3	6	8	10	15	20
S_{50}	0.083	0.088	0.078	**0.069**	0.081	0.076

Table 3.5: Simple regrets achieved by PO-GP-UCB with fixed $\epsilon = \exp(2.9)$ and different values of r after 50 iterations for the real-world loan applications dataset. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 15$.

r	3	9	12	15	20	30
S_{50}	0.091	0.076	0.078	**0.077**	0.1	0.096

Table 3.6: Simple regrets achieved by PO-GP-UCB with fixed $\epsilon = \exp(3.1)$ and different values of r after 50 iterations for the real-world loan applications dataset. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 20$.

r	5	10	15	20	30	50
S_{50}	0.097	0.091	**0.069**	0.084	0.104	0.127

3.6.3 Real-world private property price dataset

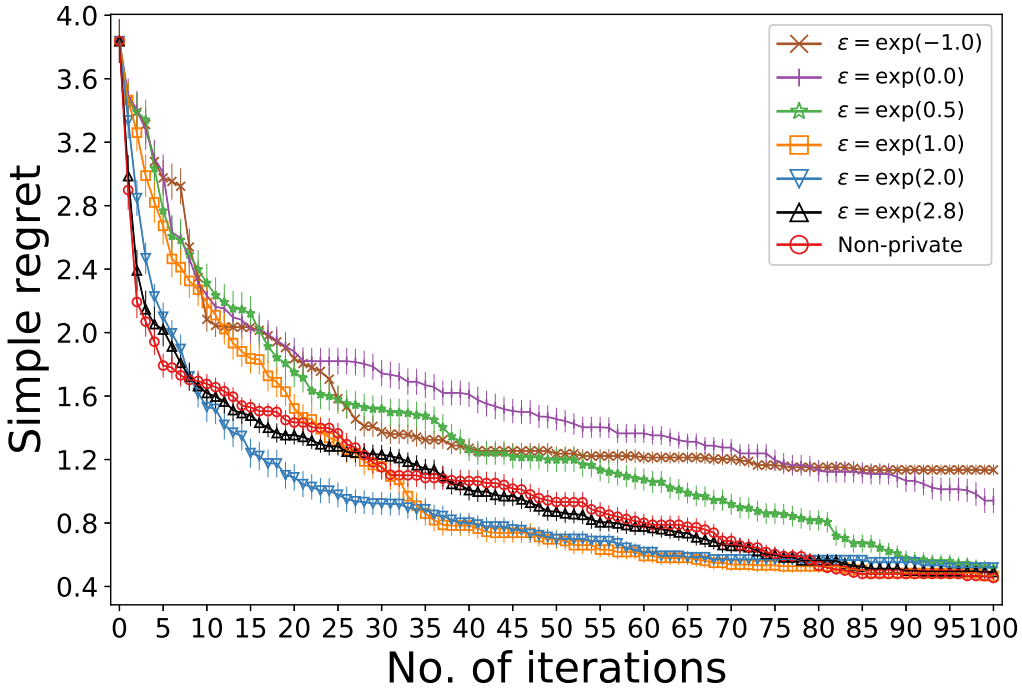


Figure 3.4: Simple regrets achieved by tested BO algorithms (with fixed r and different values of ϵ) vs. the number of iterations for private property price dataset, $r = 15$.

For this experiment we set $r = 15$ (Algorithm 2), DP parameter $\delta = 10^{-4}$ (Definition 3.5) and the GP-UCB parameter $T = 100$.

The results of this experiment for different values of ϵ are displayed in Fig. 3.4. Similar observations can be made that are consistent with the previous experiments. In particular, smaller values of ϵ (tighter privacy guarantees) generally lead to worse BO performance (larger simple regret); PO-GP-UCB with the largest value of $\epsilon = \exp(2.8)$ satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ incurs slightly larger simple regret ($0.051\sigma_y$) than non-private GP-UCB after 100 iterations; PO-GP-UCB with some values of ϵ in the single-digit range satisfying the condition $\sigma_{min}(\mathcal{X}) < \omega$ exhibits small disadvantages compared with non-private GP-UCB after 100 iterations in terms

of simple regrets: $\epsilon = \exp(1.0)$ and $\epsilon = \exp(0.5)$ result in $0.017\sigma_y$ and $0.082\sigma_y$ more simple regret respectively.

We again empirically inspect the impact of r on the performance of PO-GP-UCB in the same manner as the previous experiments: we fix the value of ϵ and vary the value of r. We consider 3 different values of DP parameter ϵ: $\epsilon = \exp(2.6)$, $\epsilon = \exp(2.8)$ and $\epsilon = \exp(3.0)$. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 10$ for $\epsilon = \exp(2.6)$, $r = 15$ for $\epsilon = \exp(2.8)$ and $r = 20$ for $\epsilon = \exp(3.0)$. Tables 3.7, 3.8 and 3.9 show that the smallest simple regret is achieved by the largest values of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$. Similar to the previous experiments, smaller values of r and larger values of r that satisfy the condition $\sigma_{min}(\mathcal{X}) < \omega$ both lead to larger simple regret, further validating the practicality of our guideline on the selection of r (Section 3.5.4).

Table 3.7: Simple regrets achieved by PO-GP-UCB with fixed $\epsilon = \exp(2.6)$ and different values of r after 100 iterations for the real-world property price dataset. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 10$.

r	3	6	8	10	15	20
S_{100}	0.682	0.516	0.495	**0.485**	**0.485**	0.493

Table 3.8: Simple regrets achieved by PO-GP-UCB with fixed $\epsilon = \exp(2.8)$ and different values of r after 100 iterations for the real-world property price dataset. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 15$.

r	3	9	12	15	20	30
S_{100}	0.567	0.553	0.479	**0.453**	0.493	0.52

Table 3.9: Simple regrets achieved by PO-GP-UCB with fixed $\epsilon = \exp(3.0)$ and different values of r after 100 iterations for the real-world property price dataset. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 20$.

r	5	10	15	20	30	50
S_{100}	0.591	0.523	0.486	**0.482**	0.489	0.488

3.6.4 Branin-Hoo benchmark function

We set the parameter $r = 10$ (Algorithm 2), DP parameter $\delta = 10^{-3}$ (Definition 3.5) and the GP-UCB parameter $T = 50$ for this experiment.

Fig. 3.5 shows the performances of PO-GP-UCB with different values of ϵ and that of non-private GP-UCB. The results are consistent with the previous experiments. Smaller values of ϵ (tighter privacy guarantees) generally lead to larger simple regret; PO-GP-UCB with the largest value of $\epsilon = \exp(2.3)$ satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ incurs only $0.004\sigma_y$ more simple regret than non-private GP-UCB after 50 iterations; PO-GP-UCB with some values of ϵ in the single-digit range satisfying the condition $\sigma_{min}(\mathcal{X}) < \omega$ exhibits small difference in simple regret compared with non-private GP-UCB after 50 iterations: $\epsilon = \exp(2.0)$ and $\epsilon = \exp(1.8)$ result in $0.023\sigma_y$ and $0.051\sigma_y$ more simple regret, respectively.

Similarly to the previous experiments, we investigate the impact of varying the value of the random projection parameter r on the performance of PO-GP-UCB. We consider 3 different values of DP parameter ϵ: $\epsilon = \exp(2.3)$, $\epsilon = \exp(2.5)$ and $\epsilon = \exp(2.7)$. We fix the value of ϵ and vary the value of r. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 10$ for $\epsilon = \exp(2.3)$, $r = 15$ for $\epsilon = \exp(2.5)$ and $r = 20$ for $\epsilon = \exp(2.7)$. Tables 3.10, 3.11 and 3.12 reveal that the largest values of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ lead to the smallest simple regret after 50 iterations. Decreasing the value of r increases the simple regret, which agrees with our analysis in Section 3.5.4 (i.e., smaller r results in worse regret upper bound). Increasing r such that the condition $\sigma_{min}(\mathcal{X}) < \omega$ is satisfied, on the other hand, also results in larger simple regret, which is again consistent with the analysis in Remark 3.2 stating that the regret upper bound becomes looser in this scenario. These observations are consisted with those for a synthetic GP dataset, a real-world loan applications dataset and a real-world property price dataset.

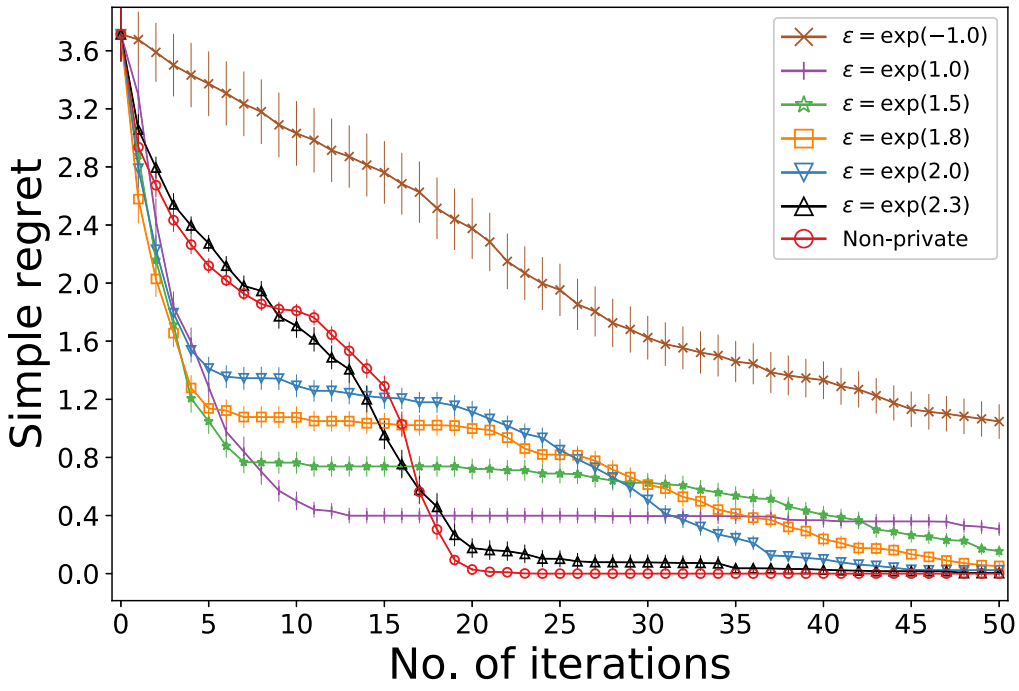


Figure 3.5: Simple regrets achieved by tested BO algorithms (with fixed $r = 10$ and different values of ϵ) vs. the number of iterations for the Branin-Hoo function dataset.

Table 3.10: Simple regrets achieved by PO-GP-UCB with fixed $\epsilon = \exp(2.3)$ and different values of r after 50 iterations for the Branin-Hoo function dataset. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 10$.

r	3	6	8	10	15	20
S_{50}	0.53	0.184	0.038	**0.0**	0.005	0.024

Table 3.11: Simple regrets achieved by PO-GP-UCB with fixed $\epsilon = \exp(2.5)$ and different values of r after 50 iterations for the Branin-Hoo function dataset. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 15$.

r	3	9	12	15	20	30
S_{50}	0.259	0.001	**0.0**	**0.0**	0.014	0.026

Table 3.12: Simple regrets achieved by PO-GP-UCB with fixed $\epsilon = \exp(2.7)$ and different values of r after 50 iterations for the Branin-Hoo function dataset. The largest value of r satisfying the condition $\sigma_{min}(\mathcal{X}) \geq \omega$ is $r = 20$.

r	5	10	15	20	30	50
S_{50}	0.152	**0.0**	**0.0**	**0.0**	0.005	0.073

Chapter 4

Nonmyopic Bayesian Optimization with Macro-Actions

This chapter of the thesis presents a principled multi-staged Bayesian sequential decision algorithm for nonmyopic adaptive BO for hotspot sampling in spatially varying phenomena. Our proposed algorithm scales up to a further lookahead (as compared to the existing nonmyopic adaptive BO algorithms [Lam *et al.*, 2016; Lam and Willcox, 2017; Ling *et al.*, 2016; Marchant *et al.*, 2014; Osborne *et al.*, 2009]) to match up to a larger available budget. To achieve this, we exploit the structure of the spatially varying phenomenon. Specifically, we rely on the notion of macro-actions (i.e., each denoting a sequence of primitive actions executed in full without considering any observation taken after performing each primitive action in the sequence) inherent to the structure of several real-world applications such as environmental sensing and monitoring, mobile sensor networks, and robotics. Some examples are given below:

- In monitoring of algal bloom in the coastal ocean, an *autonomous underwater vehicle* (AUV) is deployed on board a research vessel in search for a hotspot

of peak phytoplankton abundance and tasked to take dives from the vessel to gather "Gulper" water samples for on-deck testing that can be cast as macro-actions [Pennington *et al.*, 2016];

- In servicing the mobility demand within an urban city, an autonomous robotic vehicle in a mobility-on-demand system cruises along different road trajectories abstracted as macro-actions to find a hotspot of highest mobility demand to pick up a user [Chen *et al.*, 2015];

- In monitoring of the indoor environmental quality of an office environment [Choi *et al.*, 2012], a mobile robot mounted with a weather board is tasked to find a hotspot of peak temperature by exploring different stretches of corridors that can be naturally abstracted into macro-actions;

- In monitoring of algal bloom in the coastal ocean, an underwater glider is tasked to find a hotspot of peak chlorophyll fluorescence by optimizing its search trajectory tractably over simple ellipses of varying sizes [Leonard *et al.*, 2007] that constitute different macro-actions.

Macro-actions have in fact been well-studied and used by the planning community to scale up algorithms for planning under uncertainty to a further lookahead [He *et al.*, 2010; He *et al.*, 2011; Lim *et al.*, 2011], which is realized from a much reduced space of possible sequences of primitive actions (i.e., macro-actions) induced by the structure of the input domain/application. Additionally, macro-actions are also studied in reinforcement learning community but named as options instead [Barto and Mahadevan, 2003; Konidaris and Barto, 2007; Stolle and Precup, 2002].

In BO context, each macro-action denotes a sequence of inputs for evaluating the unknown objective function. The use of macro-actions for nonmyopic adaptive BO poses an interesting research question: *How can an acquisition function be defined*

with respect to a nonmyopic adaptive macro-action policy and optimized tractably to yield such a policy with a provable performance guarantee for a given finite budget?

The main technical difficulty in answering this question stems from the need to account for the correlation of outputs to be observed from evaluating the unknown objective function at inputs found within a macro-action and between different macro-actions. Such a correlation structure is the chief ingredient to be exploited for selecting informative observations to find the global maximum.

To design our algorithm, we first generalize GP-UCB [Srinivas *et al.*, 2010] to a new acquisition function defined with respect to a nonmyopic adaptive macro-action policy. However, an uncountable set of candidate outputs makes this policy intractable to be optimized exactly. To resolve this issue, we use stochastic sampling in each planning stage to solve for a nonmyopic adaptive ϵ-*Bayes-optimal macro-action BO* (ϵ-Macro-BO) policy given an arbitrarily user-specified loss bound ϵ and a finite budget of function evaluations, which is a key novel contribution of our work here. Additionally, we show that our proposed algorithm requires only a polynomial number of samples in the length of macro-actions[1] (Section 4.2). To perform nonmyopic adaptive BO in real time, we then propose an asymptotically optimal anytime variant of our ϵ-Macro-BO policy with a performance guarantee (Section 4.3). We use synthetic and real-world datasets to empirically evaluate the performance of our ϵ-Macro-BO policy and its anytime variant in BO (Section 4.4).

4.1 Problem setting

To simplify exposition of our work here, for the rest of this chapter we will assume the input domain $\mathcal{X}$ to be the domain of a spatially varying phenomenon (e.g., indoor

[1]In contrast, though the nonmyopic adaptive BO algorithm of [Ling *et al.*, 2016] based on deterministic sampling can be naively generalized to exploit macro-actions, it requires an exponential number of samples per planning stage (iteration), as detailed in Remark 4.3.

environmental quality of an office environment, plankton bloom in the ocean, mobility demand within an urban city, as described in the previous section). A mobile sensing agent utilizes our proposed nonmyopic adaptive ϵ-Macro-BO policy or its anytime variant to select and gather observations from the input domain for finding the global maximum. Furthermore, for the rest of this chapter we will use the term "input location" instead of "input" to match the setting of the problem. The problem setting is visually illustrated in Fig. 4.1.

To recall, let $\mathcal{X}$ be the domain of a spatially varying phenomenon corresponding to a set of input locations. In every iteration $t > 0$, the agent executes one of the available macro-actions of length κ at its current input location by deterministically moving through a sequence of κ input locations, denoted by a vector $\mathbf{x}_t \in \mathcal{A}(\mathbf{x}_{t-1})$, and observes the corresponding noisy output measurements $\mathbf{y}_t \in \mathbb{R}^\kappa$, where $\mathcal{A}(\mathbf{x}_{t-1}) \subseteq \mathcal{X}^\kappa$ denotes a finite set of available macro-actions at the agent's current input location[2] (see visual illustration in Fig. 4.1 and its caption b). The state of the agent at its initial starting input location is represented by prior observations/data $d_0 \triangleq \langle \mathbf{x}_0, \mathbf{y}_0 \rangle$ available before planning where $\mathbf{x}_0$ and $\mathbf{y}_0$ denote, respectively, vectors comprising input locations visited and corresponding output measurements observed by the agent prior to planning. The agent's initial starting input location is the last component of $\mathbf{x}_0$. In iteration $t > 0$, the state of the agent is represented by observations/data $d_t \triangleq \langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t} \rangle$ where $\mathbf{x}_{1:t} \triangleq \mathbf{x}_0 \oplus \ldots \oplus \mathbf{x}_t$ and $\mathbf{y}_{1:t} \triangleq \mathbf{y}_0 \oplus \ldots \oplus \mathbf{y}_t$ denote, respectively, vectors comprising input locations visited and corresponding output measurements observed by the agent up till iteration t and '$\oplus$' denotes vector concatenation.

The spatially varying phenomenon is modeled as a realization of a GP (Section 3.1.2): Each input location $x \in \mathcal{X}$ is associated with an output measurement $f(x)$. We assume that the covariance function $k_{xx'}$ is defined by the non-isotropic SE

[2]Note that $\mathcal{A}(\mathbf{x}_{t-1})$ depends on the agent's current input location which corresponds to the last component of macro-action $\mathbf{x}_{t-1}$ executed in the previous iteration $t-1$.

kernel (Section 3.1.2) with the signal variance σ_y^2 and length-scale components ℓ_1 and ℓ_2 controlling the spatial correlation or "similarity" between output measurements in the respective east-west and north-south directions of the 2D phenomenon[3].

Supposing the agent has gathered observations $d_t = \langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t} \rangle$ from iterations 0 to t the GP model can exploit these observations d_t to perform probabilistic regression by providing a Gaussian posterior distribution/belief of noisy output measurements for any κ input locations $\mathbf{x}_{t+1} \subset \mathcal{X}$ with the following posterior mean vector and covariance matrix, respectively [Rasmussen and Williams, 2006]:

$$\begin{aligned} \mu_{t+1}(\mathbf{x}_{t+1}) &\triangleq K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}(K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} + \sigma_n^2 I)^{-1}\mathbf{y}_{1:t}^\top \\ \Sigma_{t+1}(\mathbf{x}_{t+1}) &\triangleq K_{\mathbf{x}_{t+1}\mathbf{x}_{t+1}} + \sigma_n^2 I - K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}(K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} + \sigma_n^2 I)^{-1}K_{\mathbf{x}_{1:t}\mathbf{x}_{t+1}} \end{aligned} \tag{4.1}$$

where $K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}$ is a matrix with covariance components $k_{xx'}$ for every input x of $\mathbf{x}_{t+1}$ and x' of $\mathbf{x}_{1:t}$, $K_{\mathbf{x}_{1:t}\mathbf{x}_{t+1}}$ is the transpose of $K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}$, and $K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}}$ ($K_{\mathbf{x}_{t+1}\mathbf{x}_{t+1}}$) is a matrix with covariance components $k_{xx'}$ for every pair of inputs x, x' of $\mathbf{x}_{1:t}$ ($\mathbf{x}_{t+1}$). Note that (4.1) is very similar to (3.2) in Section 3.1.2 with the difference that in (4.1) we are interested in *simultaneous* GP prediction for κ input locations $\mathbf{x}_{t+1}$.

4.2 ϵ-Bayes-Optimal Macro-BO

To cast nonmyopic adaptive *macro-action BO* (Macro-BO) as a Bayesian sequential decision problem, we define a nonmyopic adaptive macro-action policy π to sequentially decide in each iteration t the next macro-action $\pi(d_t) \in \mathcal{A}(\mathbf{x}_t)$ to be executed for gathering κ new observations based on the current observations d_t over a finite planning horizon of H iterations (i.e., a lookahead of κH observations). The goal of the

[3]While such setting implies that we assume the dimension of inputs to be $d = 2$, this choice is only motivated by our motivating applications. All our results hold for any other input dimension d and kernel $k_{xx'}$.

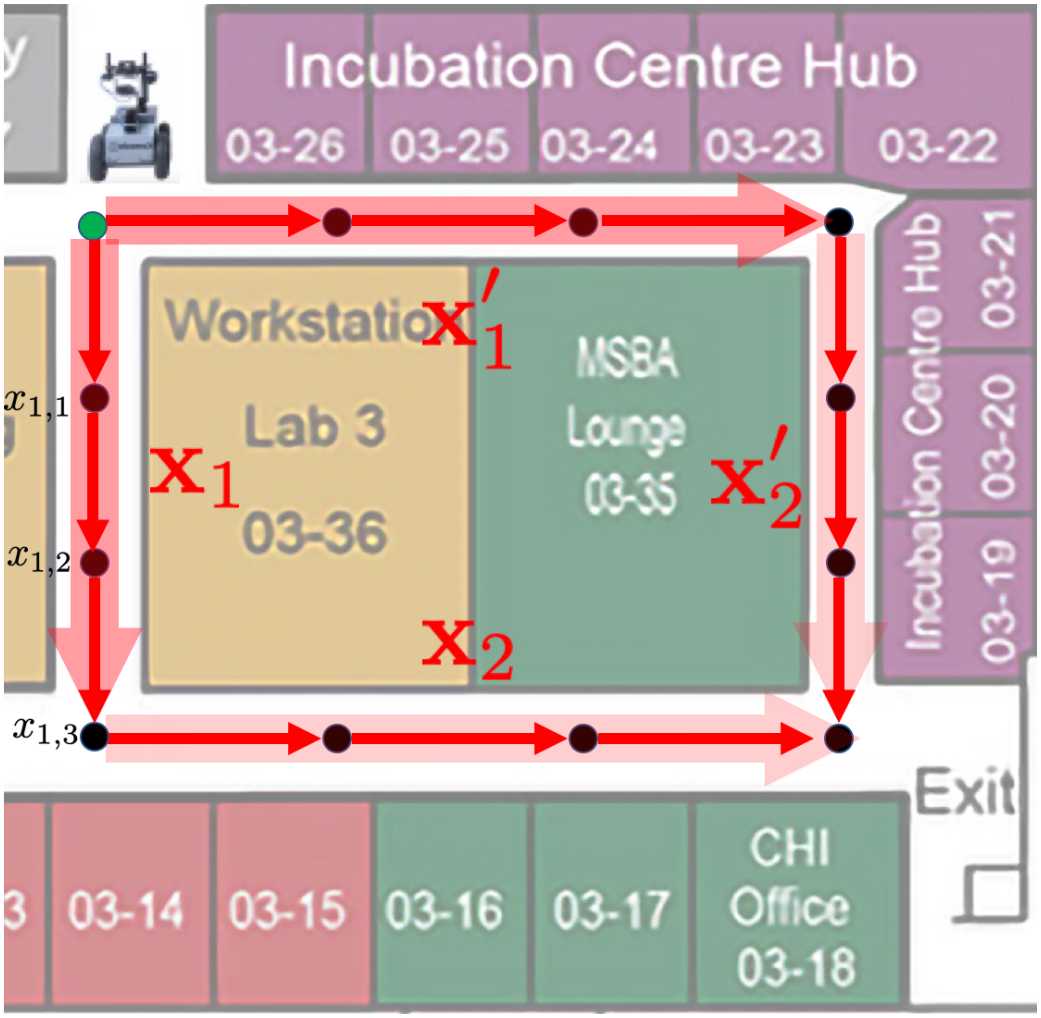


Figure 4.1: Example of monitoring indoor environmental quality of an office environment [Choi *et al.*, 2012]: (a) A mobile robot mounted with a weather board is tasked to find a hotspot of peak temperature by exploring different stretches of corridors that can be naturally abstracted into macro-actions. (b) In iteration $t = 1$, the robot is at its initial starting input location (green dot). It can decide to execute macro-action $\mathbf{x}_1$ (translucent red arrow), which is a sequence of $\kappa = 3$ primitive actions (opaque red arrows) moving it through a sequence of $\kappa = 3$ input locations (black dots) to arrive at input location $x_{1,3}$. So, $\mathbf{x}_1 \triangleq (x_{1,1}, x_{1,2}, x_{1,3})$. (c) To derive a myopic Macro-BO or ϵ-Macro-BO policy with $H = 1$, the last stages of Bellman equations in (4.5)-(4.9) require macro-actions $\mathbf{x}_1$ and $\mathbf{x}'_1$ as inputs. To derive a nonmyopic one with $H = 2$, they require macro-action sequences $\mathbf{x}_1 \oplus \mathbf{x}_2$ and $\mathbf{x}'_1 \oplus \mathbf{x}'_2$ as inputs instead.

agent is to plan/decide its macro-actions to visit input locations $\mathbf{x}_{1:H} = \mathbf{x}_1 \oplus \ldots \oplus \mathbf{x}_H$ with the maximum total corresponding output measurements

$$\mathbf{1}^\top \mathbf{y}_{1:H} = \sum_{t=1}^{H} \mathbf{1}^\top \mathbf{y}_t = \sum_{t=1}^{H} \sum_{i=1}^{\kappa} y_{t,i}$$

where $\mathbf{y}_{1:H} = \mathbf{y}_1 \oplus \ldots \oplus \mathbf{y}_H$ and $\mathbf{y}_t = (y_{t,1}, \ldots, y_{t,\kappa})$. However, since only the prior observations/data d_0 are known, the Macro-BO problem involves finding a nonmyopic adaptive macro-action policy π to select input locations $\mathbf{x}_{1:H}$ to be visited by the agent with the maximum *expected* total corresponding output measurements $\mathbb{E}_{\mathbf{y}_{1:H}|d_0,\pi}[\mathbf{1}^\top \mathbf{y}_{1:H}]$ instead.

Supposing the size of the available budget in a real-world task environment exceeds the lookahead of κH observations, it can afford a *stronger exploration behavior* by including an additional weighted exploration term $\beta\, \mathbb{I}[f(\mathcal{X}); \mathbf{y}_{1:H}|d_0, \pi]$ where $f(\mathcal{X}) \triangleq \{f(x)\}_{x\in\mathcal{X}}$. Its effect on BO performance is empirically investigated in Section 4.4. The conditional mutual information $\mathbb{I}[f(\mathcal{X}); \mathbf{y}_{1:H}|d_0, \pi]$ here can be interpreted as the information gain on the phenomenon over the entire domain $\mathcal{X}$ (i.e., equivalent to $f(\mathcal{X})$) from gathering observations $\langle \mathbf{x}_{1:H}, \mathbf{y}_{1:H}\rangle$ selected according to the nonmyopic adaptive macro-action policy π given the prior data d_0. Then, the acquisition function w.r.t. a nonmyopic adaptive macro-action policy π when starting in d_0 and following π thereafter can be defined as

$$V_0^\pi(d_0) \triangleq \mathbb{E}_{\mathbf{y}_{1:H}|d_0,\pi}[\mathbf{1}^\top \mathbf{y}_{1:H}] + \beta\, \mathbb{I}[f(\mathcal{X}); \mathbf{y}_{1:H}|d_0, \pi]\ . \tag{4.2}$$

Applying the chain rule for mutual information and a few other information-theoretic results to (4.2) yields the following H-stage Bellman equations:

$$\begin{aligned} V_t^\pi(d_t) &\triangleq Q_t^\pi(\pi(d_t), d_t)\ , \\ Q_t^\pi(\mathbf{x}_{t+1}, d_t) &\triangleq R(\mathbf{x}_{t+1}, d_t)\ + \mathbb{E}_{\mathbf{y}_{t+1}|\mathbf{x}_{t+1},d_t}[V_{t+1}^\pi(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1}\rangle)] \end{aligned} \tag{4.3}$$

for stages $t = 0, \ldots, H-1$ where $V_H^\pi(d_H) \triangleq 0$ and

$$R(\mathbf{x}_{t+1}, d_t) \triangleq \mathbf{1}^\top \mu_t(\mathbf{x}_{t+1}) + 0.5\beta \log |I + \sigma_n^{-2} \Sigma_t(\mathbf{x}_{t+1})| \ . \tag{4.4}$$

See Appendix B.1.1 for the derivation.

To solve the Macro-BO problem, Bayes-optimality[4] is exploited to select input locations to be visited by the agent that maximize the expected total corresponding output measurements (and, if the budget can afford, the additional weighted exploration term representing the information gain on the phenomenon) with respect to all possible induced sequences of future GP posterior beliefs $p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, d_t)$ for $t = 0, \ldots, H-1$. Formally, this involves choosing a nonmyopic adaptive macro-action policy π to maximize $V_0^\pi(d_0)$, which we call the Bayes-optimal Macro-BO policy π^*. That is,

$$V_0^*(d_0) \triangleq V_0^{\pi^*}(d_0) = \max_\pi V_0^\pi(d_0).$$

Plugging π^* into $V_t^\pi(d_t)$ and $Q_t^\pi(\mathbf{x}_{t+1}, d_t)$ (4.3) gives

$$\begin{aligned} V_t^*(d_t) &\triangleq \max_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} Q_t^*(\mathbf{x}_{t+1}, d_t) \ , \\ Q_t^*(\mathbf{x}_{t+1}, d_t) &\triangleq R(\mathbf{x}_{t+1}, d_t) + \mathbb{E}_{\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, d_t}[V_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle)] \end{aligned} \tag{4.5}$$

for stages $t = 0, \ldots, H-1$ where $V_H^*(d_H) \triangleq 0$.[5] When the lookahead of κH observations matches up to the available budget, the Bayes-optimal Macro-BO policy π^* can naturally trade off between exploration vs. exploitation without needing the

[4] Bayes-optimality is previously studied in discrete *Bayesian reinforcement learning* (BRL) [Poupart *et al.*, 2006] but its assumed discrete-valued output measurements and Markov property do not hold in Macro-BO. Continuous BRLs [Dallaire *et al.*, 2009; Ross *et al.*, 2008] assume a known parametric observation function, the reward function to be independent of output measurements and previous input locations, and/or, when using GP, the most likely observations during planning with no performance guarantee.

[5] To understand the effect of H on how much macro-action sequence information are required as inputs to the Bellman equations in (4.5)-(4.9), refer to Fig. 4.1 and its caption c for a visual illustration.

additional weighted exploration term in (4.2) or (4.4) (i.e., $\beta = 0$): Its selected macro-action $\pi^*(d_t) = \text{argmax}_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} \ Q_t^*(\mathbf{x}_{t+1}, d_t)$ in each iteration t has to trade off between exploiting the current GP posterior belief $p(\mathbf{y}_{t+1}|\pi^*(d_t), d_t)$ to maximize the expected total corresponding output measurements $R(\pi^*(d_t), d_t) = \mathbf{1}^\top \mu_t(\pi^*(d_t))$ vs. improving the GP posterior belief of the phenomenon (i.e., exploration) so as to maximize the expected total output measurements $\mathbb{E}_{\mathbf{y}_{t+1}|\pi^*(d_t),d_t}[V_{t+1}^*(\langle \mathbf{x}_{1:t} \oplus \pi^*(d_t), \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle)]$ in the later stages.

When the available budget is larger than the lookahead of κH observations, it can afford a *stronger exploration behavior* by setting a positive weight $\beta > 0$ on the exploration term $0.5 \log |I + \sigma_n^{-2} \Sigma_t(\pi^*(d_t))|$ in (4.4); its effect on BO performance is empirically investigated in Section 4.4. This exploration term can be interpreted as the information gain $\mathbb{I}[f(\mathcal{X}); \mathbf{y}_{t+1}|d_t, \pi^*(d_t)]]$ on the phenomenon (Appendix B.1.1) from executing the macro-action $\pi^*(d_t)$ to gather κ new observations. As such, the macro-action $\pi^*(d_t)$ can gain more information on the phenomenon (larger exploration term) by gathering observations with higher uncertainty (larger individual posterior variance) but lower correlation (smaller magnitude of posterior covariance) between them.

In general, the Macro-BO policy π^* cannot be derived exactly because the expectation term in (4.5) (and hence Q_t^* and V_t^*) often cannot be evaluated in closed form due to an uncountable set of candidate output measurements. To overcome this difficulty, we will derive a nonmyopic adaptive ϵ-Macro-BO policy π^ϵ whose expected performance loss is theoretically guaranteed to be within an arbitrarily user-specified loss bound ϵ. Preliminary to its design is the approximation of the expectation term in (4.5) for each candidate macro-action $\mathbf{x}_{t+1}$ in every iteration using *stochastic* sampling of N i.i.d. multivariate Gaussian vectors $\mathbf{y}^1, \ldots, \mathbf{y}^N$ from the GP posterior

belief $p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, d_t)$ (4.1), as illustrated in Fig. 4.2a:

$$\begin{aligned} \mathcal{V}_t(d_t) &\triangleq \max_{\mathbf{x}_{t+1}\in\mathcal{A}(\mathbf{x}_t)} \mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) \ , \\ \mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) &\triangleq R(\mathbf{x}_{t+1}, d_t) + \frac{1}{N}\sum_{\ell=1}^{N} \mathcal{V}_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell} \rangle) \end{aligned} \tag{4.6}$$

for stages $t = 0, \ldots, H-1$ where $\mathcal{V}_H(d_H) \triangleq 0$.[5] We prove in Appendix B.1.4 that $\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ (4.6) can approximate $Q_t^*(\mathbf{x}_{t+1}, d_t)$ (4.5) arbitrarily closely for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ with a high probability of at least $1-\delta$ requiring only a polynomial number N of samples in the macro-action length κ per planning stage:

Theorem 4.1. *Suppose that the observations d_t, $H \in \mathbb{Z}^+$, a budget of $\kappa(H-t)$ input locations for $t = 0, \ldots, H-1$, $\delta \in (0,1)$, and $\lambda > 0$ are given. Then, the probability of*

$$|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \leq \lambda H$$

for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ is at least $1-\delta$ by setting

$$N = \mathcal{O}((\kappa^{2H}/\lambda^2)\log(\kappa A/(\delta\lambda))) \tag{4.7}$$

where A denotes the largest number of candidate macro-actions available at any input location in $\mathcal{X}$.

Remark 4.1. Since $|\mathcal{V}_t(d_t) - V_t^*(d_t)| \leq \max_{\mathbf{x}_{t+1}\in\mathcal{A}(\mathbf{x}_t)} |\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)|$, it immediately follows from Theorem 4.1 that the probability of $|\mathcal{V}_t(d_t) - V_t^*(d_t)| \leq \lambda H$ is at least $1-\delta$.

Remark 4.2. It can be observed from Theorem 4.1 that the number N of stochastic samples increases[6] with (a) a tighter bound λ on the error $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)|$

[6]In fact, N also increases when a larger H is available and the spatial phenomenon varies with more intensity and less noise (larger σ_y^2/σ_n^2) (Appendix B.1.4). These constants are omitted from (4.10) to ease clutter.

due to stochastic sampling, (b) a higher probability $1-\delta$ of $\mathcal{Q}_t$ (4.6) approximating Q_t^* (4.5) closely, (c) a larger number A of candidate macro-actions available at any input location in $\mathcal{X}$, and (d) a greater macro-action length κ.

Deriving the above probabilistic bound usually requires using a concentration inequality involving independent Gaussian random variables. However, the components of the multivariate Gaussian random vector $\mathbf{y}_{t+1}$ in (4.5) are *correlated* output measurements corresponding to the κ input locations found within the candidate macro-action $\mathbf{x}_{t+1}$. To resolve this complication, we exploit a change of variables trick (i.e., to make the components independent) and the Lipschitz continuity of $R(\mathbf{x}_{t+1}, d_t)$ (Lemma B.1) for enabling the use of the Tsirelson-Ibragimov-Sudakov inequality [Boucheron *et al.*, 2013] to prove the probabilistic bound in Theorem 4.1, as shown in Appendix B.1.4.

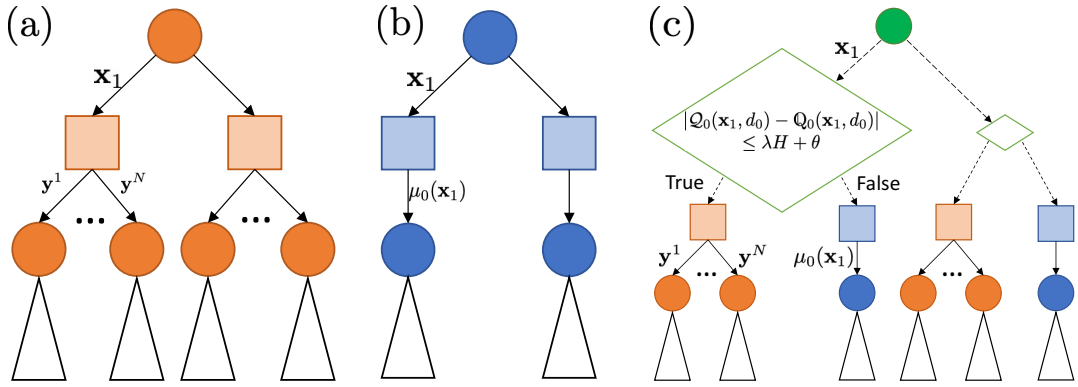


Figure 4.2: Visual illustrations of policies induced by (a) stochastic sampling (4.6), (b) most likely observations (4.8), and (c) our ϵ-Macro-BO policy π^ϵ (4.9). Circles denote nodes d_t. Squares denote nodes $\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \rangle$.

Theorem 4.1, however, only entails probabilistic bounds on how far $\mathcal{V}_t(d_t)$ (4.6) is from $V_t^*(d_t)$ (4.5) (see Remark 4.1) and on the resulting policy loss. We will prove a stronger non-trivial result: In the unlikely event (with an arbitrarily small probability of at most δ) that $\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ (4.6) is far from $Q_t^*(\mathbf{x}_{t+1}, d_t)$ (4.5) for some

$\mathbf{x}_{t+1}$, we instead rely on the κ most likely observations[7] $\mu_t(\mathbf{x}_{t+1})$ for approximating the expectation term in (4.5) (see Fig. 4.2b):

$$\begin{aligned}\mathbb{V}_t(d_t) &\triangleq \max_{\mathbf{x}_{t+1}\in\mathcal{A}(\mathbf{x}_t)} \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)\ ,\\ \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t) &\triangleq R(\mathbf{x}_{t+1}, d_t) + \mathbb{V}_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mu_t(\mathbf{x}_{t+1})\rangle)\end{aligned} \tag{4.8}$$

for stages $t = 0, \ldots, H-1$ where $\mathbb{V}_H(d_H) \triangleq 0$.[5] Unlike $\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ (4.6), the approximation quality of $\mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)$ (4.8) can be *deterministically* bounded but cannot be user-specified to be arbitrarily good, as shown in Theorem 4.2 below (see Appendix B.1.5 for the proof). To ease understanding, we visually illustrate in Fig. 4.2 how the policies induced by stochastic sampling (4.6) vs. most likely observations (4.8) differ and are used to design our ϵ-Macro-BO policy π^ϵ (4.9).

Theorem 4.2. *Suppose that the observations d_t, $H \in \mathbb{Z}^+$, and a budget of $\kappa(H-t)$ input locations for $t = 0, \ldots, H-1$ are given. Then,*

$$|\mathbb{Q}_t(\mathbf{x}_{t+1}, d_t) - Q^*_t(\mathbf{x}_{t+1}, d_t)| \leq \theta$$

for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ where $\theta \triangleq \mathcal{O}(\kappa^{H+1/2})$.

Remark 4.3. $\mathbb{V}_t$ (4.8) can be potentially generalized to resemble $\mathcal{V}_t$ (4.6) by approximating the expectation term in (4.5) for each candidate macro-action $\mathbf{x}_{t+1}$ in every stage via *deterministic* sampling from the GP posterior belief $p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, d_t) = \mathcal{N}(\mu_t(\mathbf{x}_{t+1}), \Sigma_t(\mathbf{x}_{t+1}))$ (4.1) over the κ-dimensional output measurement space of $\mathbf{y}_{t+1}$. To do this, the nonmyopic adaptive BO algorithm of [Ling *et al.*, 2016] can be extended to handle macro-actions by uniformly partitioning and sampling the κ-dimensional space of $\mathbf{y}_{t+1}$ but would consequently incur an *exponential* number of

[7] Though the nonmyopic BO algorithm of [Marchant *et al.*, 2014] assumes the most likely observations during planning, it does not consider macro-actions nor give a performance guarantee.

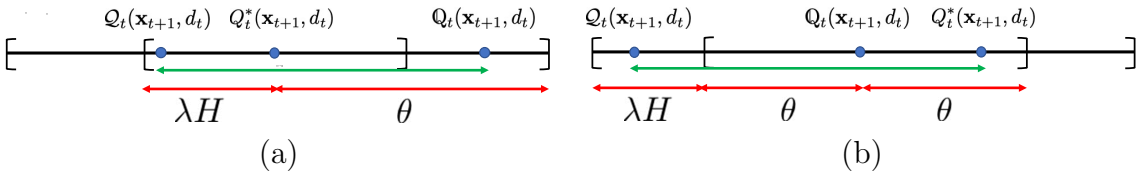


Figure 4.3: (a) When $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q^*_t(\mathbf{x}_{t+1}, d_t)| \leq \lambda H$, $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)|$ (green) is at most $\lambda H + \theta$ (red) and hence $Q^\epsilon_t(\mathbf{x}_{t+1}, d_t) = \mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$. (b) When $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q^*_t(\mathbf{x}_{t+1}, d_t)| > \lambda H$ and $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \leq \lambda H + \theta$, $Q^\epsilon_t(\mathbf{x}_{t+1}, d_t) = \mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ due to (4.9) and $|Q^\epsilon_t(\mathbf{x}_{t+1}, d_t) - Q^*_t(\mathbf{x}_{t+1}, d_t)|$ (green) is at most $\lambda H + 2\theta$ (red). All other cases (e.g., when both $\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ and $\mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)$ are larger than $Q^*_t(\mathbf{x}_{t+1}, d_t)$ in (a) or $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q^*_t(\mathbf{x}_{t+1}, d_t)| > \lambda H$ and $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| > \lambda H + \theta$ in (b), $Q^\epsilon_t(\mathbf{x}_{t+1}, d_t) = \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)$ due to (4.9)) are covered by our rigorous analysis of the implications of the if condition in (4.9) in the main text.

samples (in κ) per planning stage. In contrast, our ϵ-Macro-BO policy π^ϵ only requires a polynomial number (in κ) of samples per planning stage, as shown in Theorem 4.3.

The key question remains: Under what condition(s) should our ϵ-Macro-BO policy π^ϵ decide to follow that induced by stochastic sampling (4.6) and, if so, what is the required number N of samples in (4.6) such that its *expected* performance loss can be deterministically guaranteed to be within an arbitrarily user-specified bound ϵ? Ideally, this can be decided if we can directly assess whether $\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ (4.6) approximates $Q^*_t(\mathbf{x}_{t+1}, d_t)$ (4.5) closely (i.e., $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q^*_t(\mathbf{x}_{t+1}, d_t)| \leq \lambda H$) for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$, which unfortunately is not possible since $Q^*_t(\mathbf{x}_{t+1}, d_t)$ cannot be tractably evaluated, as explained previously. To overcome this technical difficulty, we propose a nonmyopic adaptive ϵ-Macro-BO policy π^ϵ that decides to strictly follow that induced by stochastic sampling (4.6) only if $\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ (4.6) is boundedly close

to $\mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)$ (4.8) for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$:

$$\begin{aligned} \pi^\epsilon(d_t) &\triangleq \text{argmax}_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} Q_t^\epsilon(\mathbf{x}_{t+1}, d_t) \ , \\ Q_t^\epsilon(\mathbf{x}_{t+1}, d_t) &\triangleq \begin{cases} \mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) & \text{if } |\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \\ & \quad \leq \lambda H + \theta \ , \\ \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t) & \text{otherwise;} \end{cases} \end{aligned} \tag{4.9}$$

for stages $t = 0, \ldots, H-1$.[5] Like the Bayes-optimal Macro-BO policy π^*, π^ϵ can also naturally trade off between exploration vs. exploitation, by the same reasoning as earlier. Unlike the deterministic policy π^*, π^ϵ is stochastic due to its use of stochastic sampling in $\widetilde{Q}_t$ (4.6). Of noteworthy interest and discussion are the implications of the tractable choice of the if condition in (4.9) for theoretically guaranteeing the performance of our ϵ-Macro-BO policy π^ϵ. We illustrate these implications in Fig. 4.3.

I. In the likely event (with a high probability of at least $1-\delta$) that $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \leq \lambda H$ for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ (Theorem 4.1),

$$\begin{aligned} &|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \\ &\leq |\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| + |Q_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \\ &\leq \lambda H + \theta \end{aligned}$$

for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ such that the first inequality is due to triangle inequality and the second inequality is due to Theorems 4.1 and 4.2. Consequently, according to (4.9), $Q_t^\epsilon(\mathbf{x}_{t+1}, d_t) = \mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ and $\pi^\epsilon(d_t)$ thus selects the same macro-action as the policy induced by stochastic sampling (4.6).

II. In the unlikely event (with an arbitrarily small probability of at most δ) that $\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ (4.6) is unboundedly far from $Q_t^*(\mathbf{x}_{t+1}, d_t)$ (4.5) (i.e., $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| > \lambda H$) for some $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$, $\pi^\epsilon(d_t)$ (4.9) guarantees that, for any

selected macro-action $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$,

$$
\begin{aligned}
&|Q_t^\epsilon(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \\
&= \begin{cases} |\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| & \text{if } |\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \leq \lambda H + \theta, \\ |\mathbb{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| & \text{otherwise;} \end{cases} \\
&\leq \begin{cases} |\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| + |\mathbb{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| & \text{if } |\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \\ & \leq \lambda H + \theta, \\ \theta & \text{otherwise;} \end{cases} \\
&\leq \lambda H + 2\theta \ , \quad \text{by triangle inequality and Theorem 4.2.}
\end{aligned}
$$

The above two implications of our tractable choice of the if condition in (4.9) are central to establishing our main result deterministically bounding the *expected* performance loss of π^ϵ relative to that of Bayes-optimal Macro-BO policy π^*, that is, policy π^ϵ is ϵ-Bayes-optimal.

Theorem 4.3. *Suppose that the observations d_0, $H \in \mathbb{Z}^+$, a budget of κH input locations, and a user-specified loss bound $\epsilon > 0$ are given. Then, $V_0^*(d_0) - \mathbb{E}_{\pi^\epsilon}[V_0^{\pi^\epsilon}(d_0)] \leq \epsilon$ by setting $\theta \triangleq \mathcal{O}(\kappa^{H+1/2})$ according to Theorem 4.2, $\delta = \epsilon/(8\theta H)$, and $\lambda = \epsilon/(4H^2)$ in Theorem 4.1 to yield*

$$N = \mathcal{O}((\kappa^{2H}/\epsilon^2)\log(\kappa A/\epsilon)) \tag{4.10}$$

where A denotes the largest number of candidate macro-actions available at any input location in $\mathcal{X}$.

Remark 4.4. It can be observed from Theorem 4.3 that the number N of stochastic samples increases[6] with (a) a tighter user-specified loss bound ϵ, (b) a larger number A of candidate macro-actions at any input location in $\mathcal{X}$, and (c) a greater macro-action

length κ.

4.3 Anytime ϵ-Bayes-Optimal Macro-BO

Unlike the Bayes-optimal policy π^*, our policy π^ϵ can be derived exactly since its incurred time does not depend on the size of the uncountable set of candidate output measurements. But, deriving π^ϵ (4.9) requires expanding an entire search tree of $\mathcal{O}(N^H)$ nodes to solve the H-stage Bellman equations of $\mathcal{V}_t$ (4.6), which is not always needed to achieve ϵ-Bayes optimality in practice. To ease this computational burden (e.g., for real-time planning), we propose an asymptotically optimal anytime variant of our ϵ-Macro-BO policy that can attain good BO performance quickly and improve its approximation quality over time.

The intuition behind our anytime ϵ-Macro-BO algorithm is to incrementally expand a search tree by iteratively simulating greedy exploration paths down the partially constructed tree and expanding the sub-trees rooted at nodes with the largest uncertainty of their corresponding values $V_t^*(d_t)$ so as to improve their approximation quality. Such an uncertainty at each encountered node d_t is quantified by the gap between its maintained upper and lower heuristic bounds $\overline{V}_t^*(d_t)$ and $\underline{V}_t^*(d_t)$ for the corresponding value $V_t^*(d_t)$ A new node is iteratively expanded during the execution by maximizing the gap between bounds, resulting in selecting the most uncertain regions of the state space. The bounds are then refined with the means of backpropagation from the leaves up to the root of the newly constructed sub-tree using the Lipschitz property of optimal value $V_t^*(d_t)$.

Consequently, each iteration of our anytime ϵ-Macro-BO algorithm only incurs linear time in N. The formulation of our anytime variant resembles that of ϵ-Macro-BO policy π^ϵ (4.9) except that it utilizes the lower heuristic bound instead of $\mathcal{Q}_t$ (4.6) and a modified if condition to bound its expected performance loss likewise.

4.3.1 Pseudocode

The pseudocode of our anytime ϵ-Macro-BO algorithm is presented in Algorithm 4. The essential steps of the main function Anytime-ϵ-Macro-BO are as follows:

1. Preprocessing (lines 40-42): Compute $\Sigma_t(\mathbf{x}_{t+1})$ (4.1), $L_{t+1}(\mathbf{x}_{1:t+1})$ (an auxiliary quantity defined in Definition B.1), and $\mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)$ (4.8) for all $\mathbf{x}_{1:t+1}$ reachable from $\mathbf{s}_0$ and $t = 0, \ldots, H-1$, and set θ according to Theorem 4.2;

2. Iteratively and incrementally expand the partially constructed search tree rooted at node d_0 by calling the recursive function ConstructTree (lines 44-45) so as to tighten the upper heuristic bound $\overline{V}_0^*(d_0)$ and lower heuristic bound $\underline{V}_0^*(d_0)$ of $V_0^*(d_0)$, hence reducing the gap $\omega \triangleq \overline{V}_0^*(d_0) - \underline{V}_0^*(d_0)$ (line 46); and

3. Compute our anytime $\langle \omega, \epsilon \rangle$-Macro-BO policy $\pi^{\omega\epsilon}(d_0)$ according to (4.12) (lines 47-51).

The recursive function ConstructTree traverses down the partially constructed search tree by repeatedly selecting nodes d_t with the largest uncertainty of their corresponding values $V_t^*(d_t)$ (i.e., largest gap $\overline{V}_t^*(d_t) - \underline{V}_t^*(d_t)$ between the upper and lower heuristic bounds of $V_t^*(d_t)$ so as to tighten them) until an unexplored node is reached. Specifically, if the function ConstructTree selects an explored node d_t, then the following steps are performed:

1. Choose the macro-action $\mathbf{x}_{t+1}$ with the tightest lower heuristic bound $\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t)$ of $Q_t^*(\mathbf{x}_{t+1}, d_t)$ (line 26);

2. Retrieve the samples $\{\mathbf{y}^\ell\}_{\ell=1,\ldots,N}$ previously generated by function ExpandTree at node d_t for macro-action $\mathbf{x}_{t+1}$ (line 27);

3. Recursively and incrementally expand the partially constructed sub-tree rooted at node $\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell^*} \rangle$ with the largest uncertainty of its corresponding value

$V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell^*} \rangle)$, i.e., largest gap

$$\overline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell^*} \rangle) - \underline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell^*} \rangle)$$

between the upper and lower heuristic bounds of $V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell^*} \rangle)$ so as to tighten them (lines 28-29);

4. Use the tightened upper and lower heuristic bounds of $V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell^*} \rangle)$ at node $\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell^*} \rangle$ to refine the heuristic bounds at its siblings (see Corollary 1 in Appendix B.1.7) by exploiting the Lipschitz continuity of V^*_{t+1} (Theorem B.1 in Appendix B.1.3) (line 30); and

5. Backpropagate the tightened/refined heuristic bounds at node $\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell^*} \rangle$ and its siblings to that at their parent node d_t (lines 31-35).

Otherwise, the function ConstructTree selects an unexplored node d_t and constructs a "minimal" sub-tree rooted at node d_t via the function ExpandTree (line 38), the latter of which involves the following steps:

1. For every macro-action $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$,

   (a) Draw N i.i.d. multivariate Gaussian vectors $\{\mathbf{y}^\ell\}_{\ell=1,\ldots,N}$ from GP posterior belief $p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, d_t)$ (line 5);

   (b) For every child node $\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle$, initialize the upper and lower heuris-

tic bounds of $V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell} \rangle)$ (lines 6-8) using Theorem 4.2:

$$
\begin{aligned}
&|\mathbb{V}_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell} \rangle) - V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell} \rangle)| \\
&= |\max_{\mathbf{x}_{t+2} \in \mathcal{A}(\mathbf{x}_{t+1})} \mathbb{Q}_{t+1}(\mathbf{x}_{t+2}, \langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell} \rangle) \\
&\quad - \max_{\mathbf{x}_{t+2} \in \mathcal{A}(\mathbf{x}_{t+1})} Q^*_{t+1}(\mathbf{x}_{t+2}, \langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell} \rangle)| \\
&\leq \max_{\mathbf{x}_{t+2} \in \mathcal{A}(\mathbf{x}_{t+1})} |\mathbb{Q}_{t+1}(\mathbf{x}_{t+2}, \langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell} \rangle) - Q^*_{t+1}(\mathbf{x}_{t+2}, \langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell} \rangle)| \\
&\leq \theta_{t+1}
\end{aligned}
\tag{4.11}
$$

where the equality is due to (4.5) and (4.8) and θ_{t+1} is defined in Theorem 4.2;

(c) Recursively expand/construct a "minimal" sub-tree rooted at the child node $\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\bar{\ell}} \rangle$ using the most likely sample $\mathbf{y}^{\bar{\ell}}$ (lines 9-10);

(d) Use the tightened upper heuristic bound $\overline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\bar{\ell}} \rangle)$ and lower heuristic bound $\underline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\bar{\ell}} \rangle)$ of $V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\bar{\ell}} \rangle)$ at node $\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\bar{\ell}} \rangle$ to refine the heuristic bounds at its unexplored siblings (see Corollary 1 in Appendix B.1.7) by exploiting the Lipschitz continuity of V^*_{t+1} (Theorem B.1 in Appendix B.1.3) (line 11); and

2. Backpropagate the tightened/refined heuristic bounds at node $\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\bar{\ell}} \rangle$ and its siblings to that at their parent node d_t (lines 12-16).

4.3.2 Theoretical analysis

Suppose Algorithm 4 terminates at $\omega \triangleq \overline{V}^*_0(d_0) - \underline{V}^*_0(d_0)$ (see line 46 in Algorithm 4). We will now give an anytime analogue/variant of our nonmyopic adaptive ϵ-Macro-BO

Algorithm 4 Anytime ϵ-Macro-BO

function ExpandTree(t, d_t, λ)
  if $t = H$ **then**
    return $\langle 0, 0 \rangle$
  for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ **do**
    $\{\mathbf{y}^\ell\}_{\ell=1,\dots,N} \leftarrow$ Draw N i.i.d. multivariate Gaussian vectors from GP posterior belief $p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, d_t)$ (4.1)
    for all $\mathbf{y}^\ell$ **do**
      $\underline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) \leftarrow \mathbb{V}_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) - \theta_{t+1}$ (4.11)
      $\overline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) \leftarrow \mathbb{V}_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) + \theta_{t+1}$ (4.11)
    $\bar{\ell} \leftarrow \operatorname{argmin}_{\ell \in \{1,\dots,N\}} \|\mathbf{y}^\ell - \mu_t(\mathbf{x}_{t+1})\|$
    $\langle \overline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\bar{\ell}} \rangle), \underline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\bar{\ell}} \rangle) \rangle \leftarrow$ ExpandTree$(t+1, \langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\bar{\ell}} \rangle, \lambda)$
    RefineBounds$(t, d_t, \mathbf{x}_{t+1}, \bar{\ell})$
    $R(\mathbf{x}_{t+1}, d_t) \leftarrow \mathbf{1}^\top \mu_t(\mathbf{x}_{t+1}) + 0.5\beta \log |I + \sigma_n^{-2} \Sigma_t(\mathbf{x}_{t+1})|$
    $\underline{Q}^*_t(\mathbf{x}_{t+1}, d_t) \leftarrow R(\mathbf{x}_{t+1}, d_t) + N^{-1} \sum_{\ell=1}^N \underline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) - \lambda$
    $\overline{Q}^*_t(\mathbf{x}_{t+1}, d_t) \leftarrow R(\mathbf{x}_{t+1}, d_t) + N^{-1} \sum_{\ell=1}^N \overline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) + \lambda$
  $\underline{V}^*_t(d_t) \leftarrow \max_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} \underline{Q}^*_t(\mathbf{x}_{t+1}, d_t)$
  $\overline{V}^*_t(d_t) \leftarrow \max_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} \overline{Q}^*_t(\mathbf{x}_{t+1}, d_t)$
  return $\langle \overline{V}^*_t(d_t), \underline{V}^*_t(d_t) \rangle$
function RefineBounds($t, d_t, \mathbf{x}_{t+1}, j$)
  $\{\mathbf{y}^\ell\}_{\ell=1,\dots,N} \leftarrow$ RetrieveSamples$(t, d_t, \mathbf{x}_{t+1})$
  for all $i \neq j$ **do**
    $b \leftarrow L_{t+1}(\mathbf{x}_{1:t+1}) \|\mathbf{y}^i - \mathbf{y}^j\|$
    $\underline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^i \rangle) \leftarrow \max(\underline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^i \rangle), \underline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^j \rangle) - b)$
    $\overline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^i \rangle) \leftarrow \min(\overline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^i \rangle), \overline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^j \rangle) + b)$
function ConstructTree(t, d_t, λ)
  if d_t has been explored **then**
    $\mathbf{x}_{t+1} \leftarrow \operatorname{argmax}_{\mathbf{x}'_{t+1} \in \mathcal{A}(\mathbf{x}_t)} \underline{Q}^*_t(\mathbf{x}'_{t+1}, d_t)$
    $\{\mathbf{y}^\ell\}_{\ell=1,\dots,N} \leftarrow$ RetrieveSamples$(t, d_t, \mathbf{x}_{t+1})$
    $\ell^* \leftarrow \operatorname{argmax}_{\ell \in \{1,\dots,N\}} \overline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) - \underline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle)$
    $\langle \overline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell^*} \rangle), \underline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell^*} \rangle) \rangle \leftarrow$ ConstructTree$(t+1, \langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell^*} \rangle, \lambda)$
    RefineBounds$(t, d_t, \mathbf{x}_{t+1}, \ell^*)$
    $R(\mathbf{x}_{t+1}, d_t) \leftarrow \mathbf{1}^\top \mu_t(\mathbf{x}_{t+1}) + 0.5\beta \log |I + \sigma_n^{-2} \Sigma_t(\mathbf{x}_{t+1})|$
    $\underline{Q}^*_t(\mathbf{x}_{t+1}, d_t) \leftarrow R(\mathbf{x}_{t+1}, d_t) + N^{-1} \sum_{\ell=1}^N \underline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) - \lambda$
    $\overline{Q}^*_t(\mathbf{x}_{t+1}, d_t) \leftarrow R(\mathbf{x}_{t+1}, d_t) + N^{-1} \sum_{\ell=1}^N \overline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) + \lambda$
    $\underline{V}^*_t(d_t) \leftarrow \max_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} \underline{Q}^*_t(\mathbf{x}_{t+1}, d_t)$
    $\overline{V}^*_t(d_t) \leftarrow \max_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} \overline{Q}^*_t(\mathbf{x}_{t+1}, d_t)$
    return $\langle \overline{V}^*_t(d_t), \underline{V}^*_t(d_t) \rangle$
  else
    return ExpandTree(t, d_t, λ)
function Anytime-ϵ-Macro-BO(d_0, ϵ, H)
  for all $\mathbf{x}_{1:t+1}$ reachable from $\mathbf{s}_0$ and $t = 0, \dots, H-1$ **do**
    Compute $\Sigma_t(\mathbf{x}_{t+1})$ (4.1), $L_{t+1}(\mathbf{x}_{1:t+1})$ (Definition B.1), and $\mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)$ (4.8)
  Set θ according to Theorem 4.2
  $\lambda \leftarrow 1/(4H/\epsilon + 1/(2\theta))$, $\delta \leftarrow \epsilon/(8\theta H)$
  while resources permit **do**
    $\langle \overline{V}^*_0(d_0), \underline{V}^*_0(d_0) \rangle \leftarrow$ ConstructTree$(0, d_0, \lambda)$
  $\omega \leftarrow \overline{V}^*_0(d_0) - \underline{V}^*_0(d_0)$
  for all $\mathbf{x}_1 \in \mathcal{A}(\mathbf{x}_0)$ **do**
    $Q^{\omega\epsilon}_0(\mathbf{x}_1, d_0) \leftarrow \underline{Q}^*_0(\mathbf{x}_1, d_0)$
    if $|Q^{\omega\epsilon}_0(\mathbf{x}_1, d_0) - \mathbb{Q}_0(\mathbf{x}_1, d_0)| > 2\lambda + \omega + \theta$ **then**
      $Q^{\omega\epsilon}_0(\mathbf{x}_1, d_0) \leftarrow \mathbb{Q}_0(\mathbf{x}_1, d_0)$
  return $\pi^{\omega\epsilon}(d_0) \leftarrow \operatorname{argmax}_{\mathbf{x}_1 \in \mathcal{A}(\mathbf{x}_0)} Q^{\omega\epsilon}_0(\mathbf{x}_1, d_0)$ (4.12)

policy π^{ϵ} (4.9), which we call the $\langle\omega, \epsilon\rangle$-Macro-BO policy $\pi^{\omega\epsilon}$:

$$\begin{aligned}\pi^{\omega\epsilon}(d_t) &\triangleq \operatorname{argmax}_{\mathbf{x}_{t+1}\in\mathcal{A}(\mathbf{x}_t)} Q_t^{\omega\epsilon}(\mathbf{x}_{t+1}, d_t)\\ Q_t^{\omega\epsilon}(\mathbf{x}_{t+1}, d_t) &\triangleq \begin{cases}\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t) & \text{if } \left|\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)\right| \leq 2\lambda + \omega + \theta,\\ \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t) & \text{otherwise;}\end{cases}\end{aligned} \tag{4.12}$$

for stages $t = 0, \ldots, H-1$ where $\mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)$ and θ are previously defined in (4.8) and Theorem 4.2, respectively. The implications of the tractable choice of the if condition in (4.12) for theoretically guaranteeing the performance of our $\langle\omega, \epsilon\rangle$-Macro-BO policy $\pi^{\omega\epsilon}$ as well as its theoretical analysis are similar to those of our ϵ-Macro-BO policy π^{ϵ} (4.9), and are rigorously derived in Appendix B.1.7. They result in the following theorem:

Theorem 4.4. *Suppose that the observations d_0, $H \in \mathbb{Z}^+$, a budget of κH input locations, and an arbitrarily user-specified loss bound $\epsilon > 0$ are given and Algorithm 4 terminates at $\omega \triangleq \overline{V}_0^*(d_0) - \underline{V}_0^*(d_0)$ (see line 46 in Algorithm 4). Then, $V_0^*(d_0) - \mathbb{E}_{\pi^{\omega\epsilon}}[V_0^{\pi^{\omega\epsilon}}(d_0)] \leq 2\omega H + \epsilon$ by setting θ according to Theorem 4.2, $\delta = \epsilon/(8\theta H)$ and $\lambda = 1/(4H/\epsilon + 1/(2\theta))$ in Theorem 4.1 to yield*

$$N = \mathcal{O}\left(\frac{\kappa^{2H}}{\epsilon^2}\log\frac{\kappa A}{\epsilon}\right).$$

4.4 Experimental results

This section empirically evaluates the performance of our nonmyopic adaptive ϵ-Macro-BO policy and its anytime variant for a given finite budget with three datasets:

- *Simulated plankton density phenomena.* An *autonomous underwater vehicle* (AUV) is deployed on board of a *research vessel* (RV) in search for a hotspot

of peak phytoplankton abundance (i.e., algal bloom) in coastal ocean. The AUV and RV are initially positioned near the center of the plankton density (mg/m^3) phenomenon spatially distributed over a 5 km by 5 km region that is discretized into a 50×50 grid of input locations. The phenomenon is modeled as a realization of a GP and simulated using the GP hyperparameters $\mu_s = 0$, $\ell_1 = \ell_2 = 0.5$ km, $\sigma_y^2 = 1$, and $\sigma_n^2 = 10^{-5}$. The AUV is tasked to execute the selected macro-action of a straight dive (due to limited maneuverability) along one of the 4 cardinal directions from the RV to gather "Gulper" water samples/observations over $\kappa = 4$ input locations for precise on-deck testing [Pennington *et al.*, 2016]; given a budget of 20 observations, this is repeated for 5 times (i.e. 5 iterations) from the input location that it has previously surfaced.

- *Real-world traffic phenomenon.* To service the mobility demands within the central business district of an urban city, an *autonomous vehicle* (AV) in a mobility-on-demand system cruises along different road trajectories to find a hotspot of highest mobility demand to pick up a user. The 29.4 km by 11.9 km service area is gridded into 100×50 input regions, of which only 2506 input regions are accessible to the AV via the road network. The AV can cruise from input region s to an adjacent input region s' using one primitive action iff at least one road segment in the road network starts in s and ends in s'; the maximum outdegree from any input region is 8. In any input region, a surrogate demand measurement is obtained by counting the number of pickups[8] from all historic taxi trajectories generated by a major taxi company during 9:30-10 p.m. on August 2, 2010 [Chen *et al.*, 2015]; the resulting mobility demand pattern is

[8] A distributed gossip-based protocol can be used to aggregate these pickup information from the AVs in the input region that are connected via an ad hoc wireless communication network [Chen *et al.*, 2015]. Any AV entering the input region can then access its pickup count by joining its ad hoc network.

visualized in Fig.4.4. The original demand measurements are log-transformed to remove skewness and extremity for stabilizing the GP covariance structure and the GP hyperparameters $\mu_s = 1.5673$, $\ell_1 = 0.1689$ km, $\ell_2 = 0.1275$ km, $\sigma_y^2 = 0.7486$, and $\sigma_n^2 = 0.0111$ are then learned using maximum likelihood estimation [Rasmussen and Williams, 2006]; note that the length-scales and signal-to-noise ratio are relatively smaller than that of the simulated plankton density phenomena. The AV is tasked to execute the selected macro-action of a cruising trajectory along $\kappa = 5$ adjacent input regions to observe their corresponding demand measurements; given a budget of 20 observations, this will be repeated for 4 times from the input region that it has previously cruised to. Since every input region s has a large number of available macro-actions (i.e., with an average of 178 and maximum of 1193 macro-actions), 20 of them are randomly[9] selected to form its representative set of candidate macro-actions.

- *Real-world temperature phenomenon.* In monitoring of the indoor environmental quality of an office environment [Choi *et al.*, 2012], a mobile robot mounted with a weather board is tasked to find a hotspot of peak temperature by exploring different stretches of corridors that can be naturally abstracted into macro-actions. The temperature (°C) phenomenon is spatially distributed over the Intel Berkeley Research Lab (of about 41 m by 32 m in size) with 41 deployed temperature sensors (see Fig. 4.5) and modeled as a realization of a GP. Using the observations/data gathered by the 41 temperature sensors[10], the GP hyperparameters $\mu_s = 17.8513$, $\ell_1 = 4.0058$ m, $\ell_2 = 11.3811$ m, $\sigma_y^2 = 0.5964$, and $\sigma_n^2 = 0.0597$ are learned using maximum likelihood estimation [Rasmussen and Williams, 2006]. Then, using these learned hyperparameters and the observa-

[9]The BO performance of ϵ-Macro-BO and its anytime variant can be potentially improved by using macro-action generation algorithms [He *et al.*, 2011] instead of random selection.

[10]http://db.csail.mit.edu/labdata/labdata.html

tions/data gathered by the 41 temperature sensors, we exploit the GP posterior mean (3.2) to predict the temperature measurements at the 104 input locations shown in Fig. 4.5; these predictions together with the data obtained from the 41 sensors serve as the dataset for the experiment here. The mobile robot is tasked to execute the selected macro-action of a motion path along a stretch of $\kappa = 5$ input locations on one of the corridors in the lab to observe their corresponding temperature measurements; given a budget of 20 observations, this will be repeated for 4 times from the input location that it has previously moved to. Since every input location s has a large number of available macro-actions (i.e., with an average of 27 and maximum of 114 macro-actions), 20 of them are randomly[9] selected to form its representative set of candidate macro-actions.

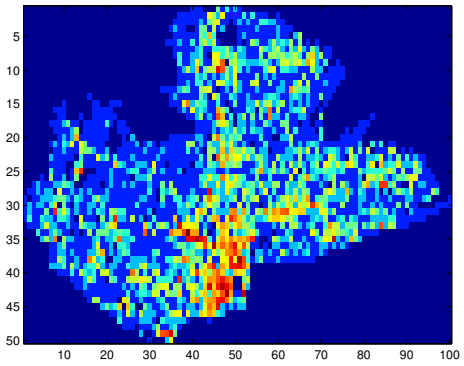


Figure 4.4: Mobility demand pattern spatially distributed over the central business district of an urban city during 9:30-10 p.m. on August 2, 2010: "Hotter" regions indicate larger numbers of pickups (Image courtesy of [Chen *et al.*, 2015]).

The performances of our ϵ-Macro-BO policy and its anytime variant are compared with that of state-of-the-art (a) nonmyopic GP-UCB [Marchant *et al.*, 2014] generalized to handle macro-actions that coincides with our deterministic policy (4.8) exploiting the most likely observations during planning, (b) *distributed batch GP-UCB* (DB-GP-UCB) [Daxberger and Low, 2017] that casts a macro-action as a batch to

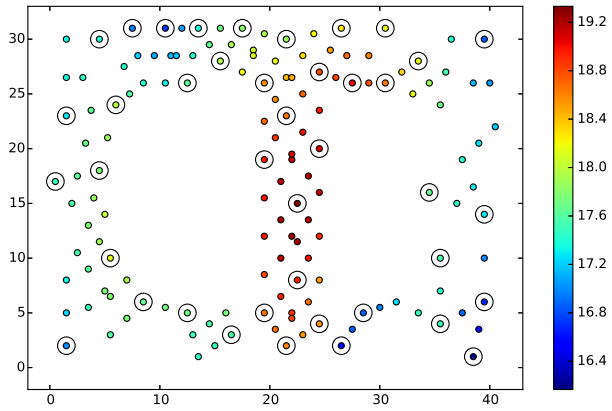


Figure 4.5: The temperature measurements at the 104 input locations (not circled) in the Intel Berkeley Research lab are predicted using the GP posterior mean (3.2) based on the data gathered by the 41 temperature sensors (circled); these predictions together with the data obtained from the 41 sensors serve as the dataset for the experiment here.

be optimized and is thus equivalent to ϵ-Macro-BO with $H = 1$, (c) q-EI [Chevalier and Ginsbourger, 2013] that does likewise, and (d) greedy batch BO algorithms[11] such as GP-BUCB [Desautels *et al.*, 2014], GP-UCB-PE [Contal *et al.*, 2013], and BBO-LP [González *et al.*, 2016a] whose implementations are detailed in Table 4.1. It is not obvious to us how GLASSES [González *et al.*, 2016b] and Rollout [Lam *et al.*, 2016] can be modified to handle macro-actions and are thus not empirically compared here. However, since Rollout [Lam *et al.*, 2016] also exploits Bellman equations, it is compared with our ϵ-Macro-BO in Section 4.4.4 by setting macro-action length to $\kappa = 1$ (i.e., primitive action).

Four performance metrics are used: (a) average normalized[12] output measurements observed by the agent (larger average output measurements imply less average/cumulative regret), (b) simple regret, (c) no. of explored nodes in all constructed

[11]Unlike DB-GP-UCB and q-EI, a greedy batch BO algorithm cannot exploit the full informativeness of any candidate macro-action for its macro-action selection: Since it selects the inputs of a batch one at a time myopically, its first few selected input locations immediately decide its chosen macro-action and consequently the remaining sequence of input locations found within.

[12]To ease interpretation of results, the prior mean is subtracted from each output measurement to normalize it.

Table 4.1: Details on the available implementations of the batch BO algorithms for comparison with ϵ-Macro-BO in our experiments.

BO Algorithm	Language	URL of Source Code
GP-BUCB	MATLAB	http://www.gatsby.ucl.ac.uk/~tdesautels/
GP-UCB-PE	MATLAB	http://econtal.perso.math.cnrs.fr/software/
q-EI	Python	https://github.com/oxfordcontrol/Bayesian-Optimization
BBO-LP	Python	http://sheffieldml.github.io/GPyOpt/

search trees (more nodes incur more time), and (d) average runtime per iteration.

4.4.1 Simulated plankton density phenomena

Figs. 4.6a and 4.6b show results of the performances of ϵ-Macro-BO with $H = 2, 3, 4$ (lookahead of, respectively, 8, 12, 16 observations), $\beta = 0$, and $N = 100$,[13] and the other tested BO algorithms averaged over 250 independent realizations of the simulated phenomena. It can be observed that as the number of observations increases, the nonmyopic adaptive BO algorithms generally outperform the myopic ones. In particular, the performance of ϵ-Macro-BO improves considerably by increasing H: ϵ-Macro-BO with the furthest lookahead (i.e., $H = 4$) achieves the largest average normalized output measurements observed by the AUV and smallest simple regret after 20 observations at the cost of a larger number of explored nodes (see Table 4.2). For example, the nonmyopic ϵ-Macro-BO with $H = 4$ achieves $0.093\sigma_y$ ($0.059\sigma_y$) more average output measurements and $0.211\sigma_y$ ($0.148\sigma_y$) less simple regret than myopic DB-GP-UCB (nonmyopic GP-UCB with the same horizon $H = 4$ but assuming most likely observations during planning), which are expected.

Figs. 4.6c and 4.6d show the effect of varying exploration weights β on the performance of ϵ-Macro-BO with $H = 2$ and $H = 3$, respectively. It can be observed from Fig. 4.6c that when $H = 2$, ϵ-Macro-BO with $\beta = 0.1$ achieves $0.064\sigma_y$ more average normalized output measurements than that with $\beta = 0$ after 20 observations,

[13]Specifying the value of N (instead of ϵ) may yield a loose ϵ based on Theorem 4.3. Nevertheless, the resulting ϵ-Macro-BO with $H = 3$,4 empirically outperforms other tested BO algorithms.

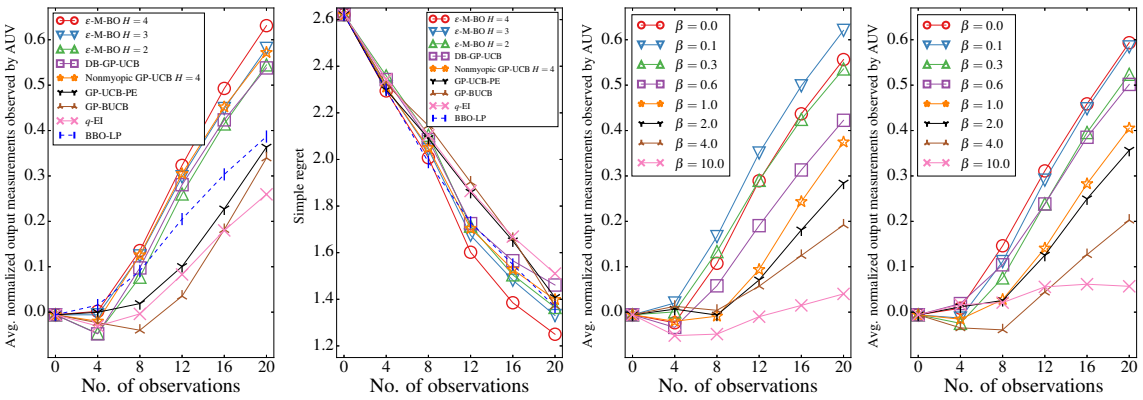


Figure 4.6: Graphs of (a) average normalized[12] output measurements observed by AUV, (b) simple regrets achieved by tested BO algorithms, average normalized output measurements achieved by ϵ-Macro-BO (ϵ-M-BO in the graphs) with (c) $H = 2$ and (d) $H = 3$ and varying exploration weights β vs. no. of observations for simulated plankton density phenomena. Standard errors are given in Tables B.1 and B.2 in Appendix B.2.1.

which indicates the need of a slightly stronger exploration behavior. Fig. 4.6d shows that by increasing to a lookahead of 12 observations (i.e., $H = 3$), ϵ-Macro-BO no longer needs the additional weighted exploration term in (4.4) (i.e., $\beta = 0$) since it can naturally trade off between exploration vs. exploitation, as explained previously (Section 4.2). It can also be observed from Figs. 4.6c and 4.6d that $\beta = 10$ greatly hurts its performance due to an overly aggressive exploration.

Table 4.2: No. of explored nodes by ϵ-Macro-BO (when $H = 1$, it corresponds to DB-GP-UCB) for simulated plankton density phenomena.

$H = 1$	$H = 2$	$H = 3$	$H = 4$
2.50×10	8.01×10^3	2.40×10^6	6.41×10^8

4.4.2 Real-world traffic phenomenon

Figs. 4.7a and 4.7b show results of the performances of *anytime* ϵ-Macro-BO with $H = 2, 3, 4$ (a lookahead of, respectively, 10, 15, 20 observations), $\beta = 0$, and $N = 300$ after

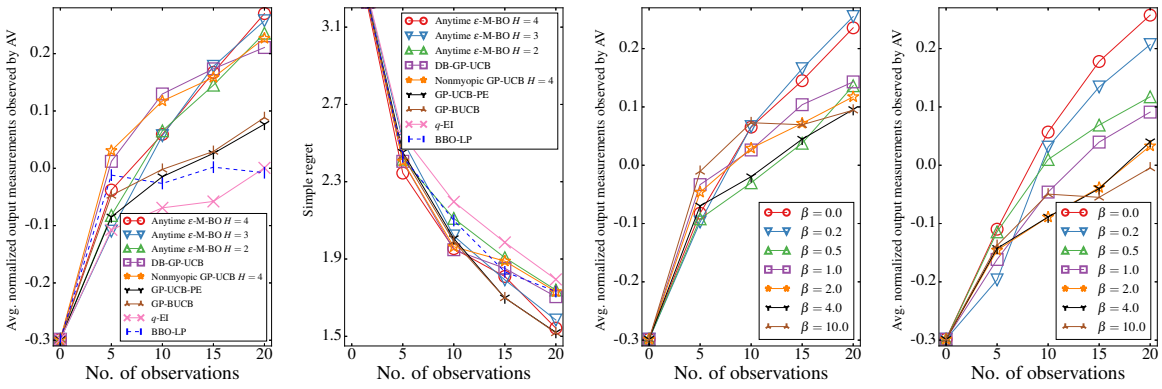


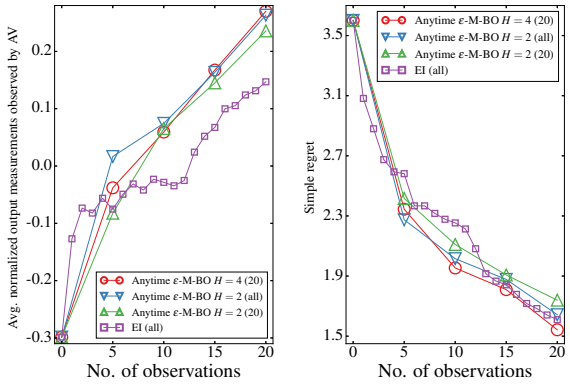
Figure 4.7: Graphs of (a) average normalized[12] output measurements observed by the AV and (b) simple regrets achieved by the tested BO algorithms, and average normalized output measurements achieved by *anytime* ϵ-Macro-BO with (c) $H = 2$ and (d) $H = 3$ and varying exploration weights β vs. no. of observations for real-world traffic phenomenon. The standard errors are given in Tables B.3 and B.4 in Appendix B.2.2.

running for 1500 iterations[13], and the other tested BO algorithms averaged over 35 random starting input regions of the AV. Similar to the results for simulated plankton density phenomena, it can be observed that the performance of anytime ϵ-Macro-BO improves considerably by increasing H: Anytime ϵ-Macro-BO with the furthest lookahead (i.e., $H = 4$) achieves the largest average normalized output measurements observed by the AV and among the least simple regret after 20 observations at the cost of a larger number of explored nodes (see Table 4.3). For example, the nonmyopic anytime ϵ-Macro-BO with $H = 4$ achieves $0.069\sigma_y$ ($0.05\sigma_y$) more average output measurements and $0.188\sigma_y$ ($0.219\sigma_y$) less simple regret than myopic DB-GP-UCB (nonmyopic GP-UCB with $H = 4$), which are expected. Interestingly, GP-BUCB and GP-UCB-PE can achieve simple regret comparable to that of anytime ϵ-Macro-BO with $H = 4$ even though they perform very poorly in terms of average output measurements.

Figs. 4.7c and 4.7d show the effect of varying exploration weights β on the per-

Table 4.3: No. of explored nodes by anytime ϵ-Macro-BO (when $H = 1$, it corresponds to DB-GP-UCB) for the real-world traffic phenomenon (i.e., mobility demand pattern).

$H = 1$	$H = 2$	$H = 3$	$H = 4$
8.29×10	9.52×10^4	1.29×10^6	1.34×10^7



(a) (b)

Figure 4.8: Graphs of (a) average normalized output measurements observed by the AV and (b) simple regrets achieved by *anytime* ϵ-Macro-BO with $H = 2, 4$ and 20 randomly selected macro-actions per input region, anytime ϵ-Macro-BO with $H = 2$ and all available macro-actions (the no. of available macro-actions per input region is enclosed in brackets), and EI with all available macro-actions of length 1 vs. no. of observations for real-world traffic phenomenon. Standard errors are given in Table B.5 in Appendix B.2.2.

formance of anytime ϵ-Macro-BO with $H = 2$ and $H = 3$, respectively. It can be observed from Fig. 4.7c that when $H = 2$, anytime ϵ-Macro-BO with $\beta = 0.2$ achieves $0.022\sigma_y$ more average normalized output measurements than that with $\beta = 0$ after 20 observations, which indicates the need of a slightly stronger exploration behavior. Fig. 4.7d shows that by increasing to a lookahead of 15 observations(i.e., $H = 3$), anytime ϵ-Macro-BO no longer needs the additional weighted exploration term in (4.4) (i.e., $\beta = 0$) since it can naturally trade off between exploration vs. exploitation, as explained previously (Section 4.2). It can also be observed from Figs. 4.7c and 4.7d that $\beta \geq 0.5$ hurts its performance due to overly aggressive exploration.

Lastly, we investigate the effect of downsampling the number of available macro-

Table 4.4: No. of explored nodes by anytime ϵ-Macro-BO (the no. of available macro-actions per input region is enclosed in brackets) for the real-world traffic phenomenon (i.e., mobility demand pattern).

$H = 2$ (20)	$H = 2$ (all)	$H = 4$ (20)
0.95×10^5	1.26×10^6	1.34×10^7

actions per input region to 20 on the performance of anytime ϵ-Macro-BO. To do this, the performances of anytime ϵ-Macro-BO with $H = 2, 4$ and 20 randomly selected macro-actions per input region are compared with that of anytime ϵ-Macro-BO with $H = 2$ and all available macro-actions as well as myopic EI [Shahriari *et al.*, 2016] with all available macro-actions of length 1. It can be observed from Figs. 4.8a and 4.8b that when $H = 2$, downsampling the number of available macro-actions per input region to 20 decreases average normalized output measurements by $0.032\sigma_y$ and increases simple regret by $0.112\sigma_y$ after 20 observations, but also reduces the number of explored nodes by more than 1 order of magnitude (see Table 4.4). By increasing to a lookahead of 20 observations, anytime ϵ-Macro-BO with $H = 4$ and 20 randomly selected macro-actions per input region achieves $0.008\sigma_y$ more average normalized output measurements and $0.116\sigma_y$ less simple regret than that with $H = 2$ and all available macro-actions at the cost of a larger number of explored nodes. Though EI can access all available macro-actions of length 1 (i.e, no restriction on action space of AV), it obtains much less average normalized output measurements and more simple regret than anytime ϵ-Macro-BO with $H = 4$ and 20 randomly selected macro-actions per input region due to its myopia.

4.4.3 Real-world temperature phenomenon

Figs. 4.9a and 4.9b show results of the performances of *anytime* ϵ-Macro-BO with $H = 2, 3, 4$ (lookahead of, respectively, 10, 15, 20 observations), $\beta = 0$, and $N = 300$ after running for 1500 iterations[13], and the other tested BO algorithms averaged over

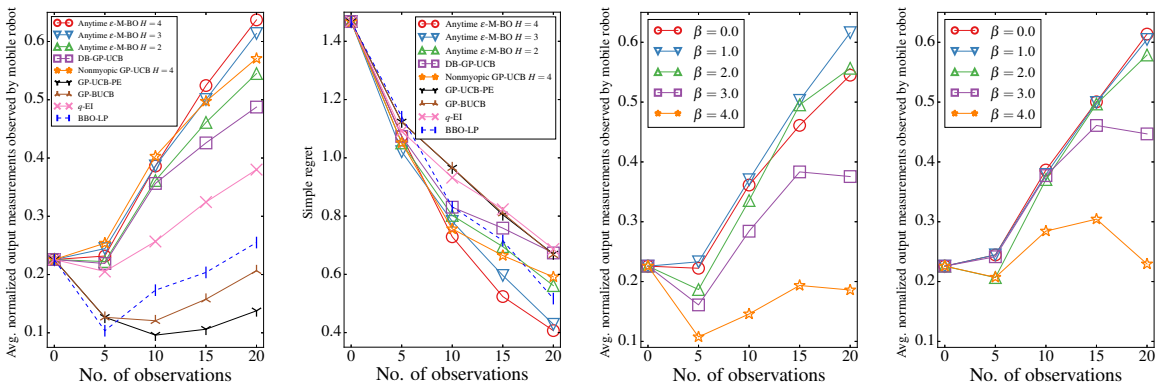


Figure 4.9: Graphs of (a) average normalized[12] output measurements observed by the mobile robot and (b) simple regrets achieved by the tested BO algorithms vs. no. of observations, and average normalized output measurements achieved by *anytime* ϵ-Macro-BO with (c) $H = 2$ and (d) $H = 3$ and varying exploration weights β vs. no. of observations for the real-world temperature phenomenon over the Intel Berkeley Research Lab. The standard errors are given in Tables B.6 and B.7 in Appendix B.2.3.

35 random initial starting input locations of the mobile robot. Similar to the results for simulated plankton density phenomena and real-world traffic phenomenon, it can be observed that as the number of observations increases, the nonmyopic adaptive BO algorithms generally outperform the myopic ones. In particular, the performance of anytime ϵ-Macro-BO improves considerably by increasing H such that anytime ϵ-Macro-BO with the furthest lookahead (i.e., $H = 4$) achieves the largest average normalized output measurements observed by the mobile robot and smallest simple regret after 20 observations at the cost of a larger number of explored nodes (see Table 4.5). For example, the nonmyopic anytime ϵ-Macro-BO with $H = 4$ achieves $0.194\sigma_y$ ($0.086\sigma_y$) more average normalized output measurements and $0.345\sigma_y$ ($0.239\sigma_y$) less simple regret than the myopic DB-GP-UCB (nonmyopic GP-UCB with the same horizon $H = 4$ but assuming most likely observations during planning), which are expected.

Figs. 4.9c and 4.9d show the effect of varying exploration weights β on the per-

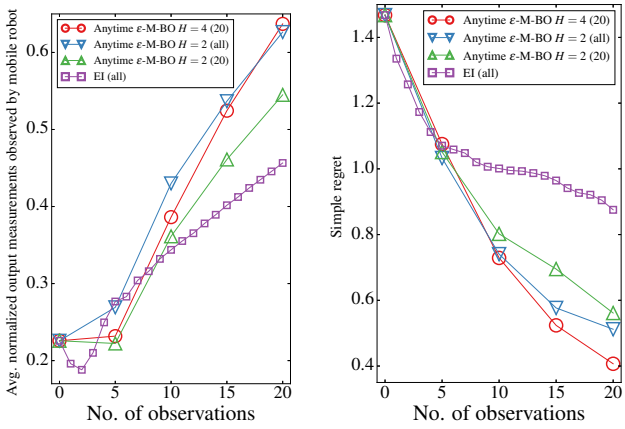
Table 4.5: No. of explored nodes by anytime ϵ-Macro-BO (when $H = 1$, it corresponds to DB-GP-UCB) for the real-world temperature phenomenon over the Intel Berkeley Research Lab.

$H = 1$	$H = 2$	$H = 3$	$H = 4$
7.51×10	8.88×10^4	1.13×10^6	1.12×10^7

formance of anytime ϵ-Macro-BO with $H = 2$ and $H = 3$, respectively. It can be observed from Fig. 4.9c that when $H = 2$, anytime ϵ-Macro-BO with $\beta = 1$ achieves $0.092\sigma_y$ more average normalized output measurements than that with $\beta = 0$ after 20 observations, which indicates the need of a slightly stronger exploration behavior. Fig. 4.9d shows that by increasing to a lookahead of 15 observations (i.e., $H = 3$), anytime ϵ-Macro-BO no longer needs the additional weighted exploration term in (4.4) (i.e., $\beta = 0$) since it can naturally trade off between exploration vs. exploitation, as explained previously (Section 4.2). It can also be observed from Figs. 4.9c and 4.9d that $\beta \geq 3$ hurts its performance due to overly aggressive exploration.

Lastly, we investigate the effect of downsampling the number of available macro-actions per input location to 20 on the performance of anytime ϵ-Macro-BO. Similar to that for the real-world traffic phenomenon, the performances of anytime ϵ-Macro-BO with $H = 2, 4$ and 20 randomly selected macro-actions per input location are compared with that of anytime ϵ-Macro-BO with $H = 2$ and all available macro-actions as well as myopic EI [Shahriari *et al.*, 2016] with all available macro-actions of length 1. It can be observed from Figs. 4.10a and 4.10b that when $H = 2$, downsampling the number of available macro-actions per input location to 20 decreases average normalized output measurements by $0.106\sigma_y$ and increases simple regret by $0.064\sigma_y$ after 20 observations, but also reduces the number of explored nodes (see Table 4.6). By increasing to a lookahead of 20 observations, anytime ϵ-Macro-BO with $H = 4$ and 20 randomly selected macro-actions per input location achieves average normalized output measurements comparable to that with $H = 2$ and all

available macro-actions, but $0.136\sigma_y$ less simple regret at the cost of a larger number of explored nodes. Though EI can access all available macro-actions of length 1 (i.e, no restriction on action space of the mobile robot), it obtains much less average normalized output measurements and considerably more simple regret than anytime ϵ-Macro-BO with $H = 4$ and 20 randomly selected macro-actions per input location due to its myopia.



(a) (b)

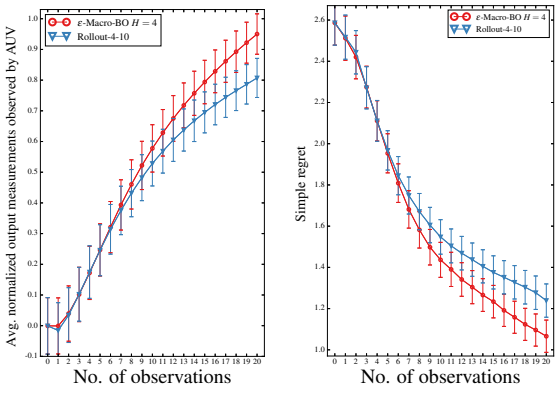
Figure 4.10: Graphs of (a) average normalized[12] output measurements observed by the mobile robot and (b) simple regrets achieved by *anytime* ϵ-Macro-BO with $H = 2, 4$ and 20 randomly selected macro-actions per input region, anytime ϵ-Macro-BO with $H = 2$ and all available macro-actions (the no. of available macro-actions per input region is enclosed in brackets), and EI with all available macro-actions of length 1 vs. no. of observations for the real-world temperature phenomenon over the Intel Berkeley Research Lab. The standard errors are given in Table B.8 in Appendix B.2.3.

Table 4.6: No. of explored nodes by anytime ϵ-Macro-BO (the no. of available macro-actions per input region is enclosed in brackets) for the real-world temperature phenomenon over the Intel Berkeley Research Lab.

$H = 2$ (20)	$H = 2$ (all)	$H = 4$ (20)
8.88×10^4	2.49×10^5	1.12×10^7

4.4.4 Comparison with Rollout [Lam et al., 2016]

Our proposed algorithms are not benchmarked against Rollout [Lam *et al.*, 2016] because Rollout [Lam *et al.*, 2016] is not designed to handle macro-actions that are inherent to the structure of the task environments/applications considered in our work and experiments. So, such a comparison would not be fair. For a fair comparison with Rollout [Lam *et al.*, 2016], we set the macro-action length to $\kappa = 1$ (i.e., primitive action) for our ϵ-Macro-BO and evaluate their performances using the metrics of average normalized output measurements observed by the agent and simple regret, and the synthetic dataset featuring the simulated plankton density phenomena in Section 4.4.



(a) (b)

Figure 4.11: Graphs of (a) average normalized[12] output measurements observed by AUV and (b) simple regrets achieved by ϵ-Macro-BO with $H = 4$ and Rollout-4-10 vs. no. of observations for simulated plankton density phenomena (Section 4.4).

Figs. 4.11a and 4.11b show results of the performances of ϵ-Macro-BO ($H = 4$, $\beta = 0$, and $N = 20$) and the best-performing Rollout ($H = 4$, $\gamma = 1.0$, base policy: greedy EI-based policy defined in equations 22 and 23 in [Lam *et al.*, 2016]) reported on page 7 in [Lam *et al.*, 2016] averaged over 106 independent realizations of the simulated phenomena. It can be observed that ϵ-Macro-BO achieves $0.143\sigma_y$ more average

normalized output measurement and $0.173\sigma_y$ less simple regret than Rollout [Lam *et al.*, 2016]. To explain this, ϵ-Macro-BO considers all available actions from each input location during planning (equations 4.6, 4.8, and 4.9) while Rollout utilizes only the action selected by the base policy (e.g., greedy EI) and ignores all the other available actions during planning, thus resulting in its suboptimal behavior.

4.4.5 Behavior of a myopic vs. nonmyopic method

In this section we illustrate the difference in behaviors of a nonmyopic vs. myopic method using our nonmyopic ϵ-Macro-BO policy with a lookahead of 8 observations ($H = 4$, $N = 1$) in Fig. 4.12a vs. greedy/myopic DB-GP-UCB [Daxberger and Low, 2017] in Fig. 4.12c with macro-action length $\kappa = 2$ and budget of 20 observations. We use the setting of controlling an AUV to gather observations for finding a hotspot (i.e., global maximum) in a simulated plankton density phenomenon (Section 4.4). Prior observations are at the AUV's initial starting input location (blue circle) and buoy's location $(0, 0)$ (not shown here).

Up till $t = 5$, both ϵ-Macro-BO and DB-GP-UCB produce the same trajectories to reach the input location denoted by a black circle. At $t = 5$, since ϵ-Macro-BO is able to look ahead and plan its macro-actions in the later planning stages, it moves the AUV left to reach the region containing the global maximum (Fig. 4.12a). On the other hand, DB-GP-UCB moves the AUV right towards the local maximum (Fig. 4.12c).

This behavior is further explained in Fig. 4.12b and Fig. 4.12d. These figures plot maps of GP posterior mean (3.2) over the phenomenon at iteration $t = 5$. The maps are identical for both algorithms, since till $t = 5$, both ϵ-Macro-BO and DB-GP-UCB produce the same trajectories. The green arrow in Fig. 4.12b denotes the macro-action selected by ϵ-Macro-BO at iteration $t = 5$, while the red arrows denote the macro-

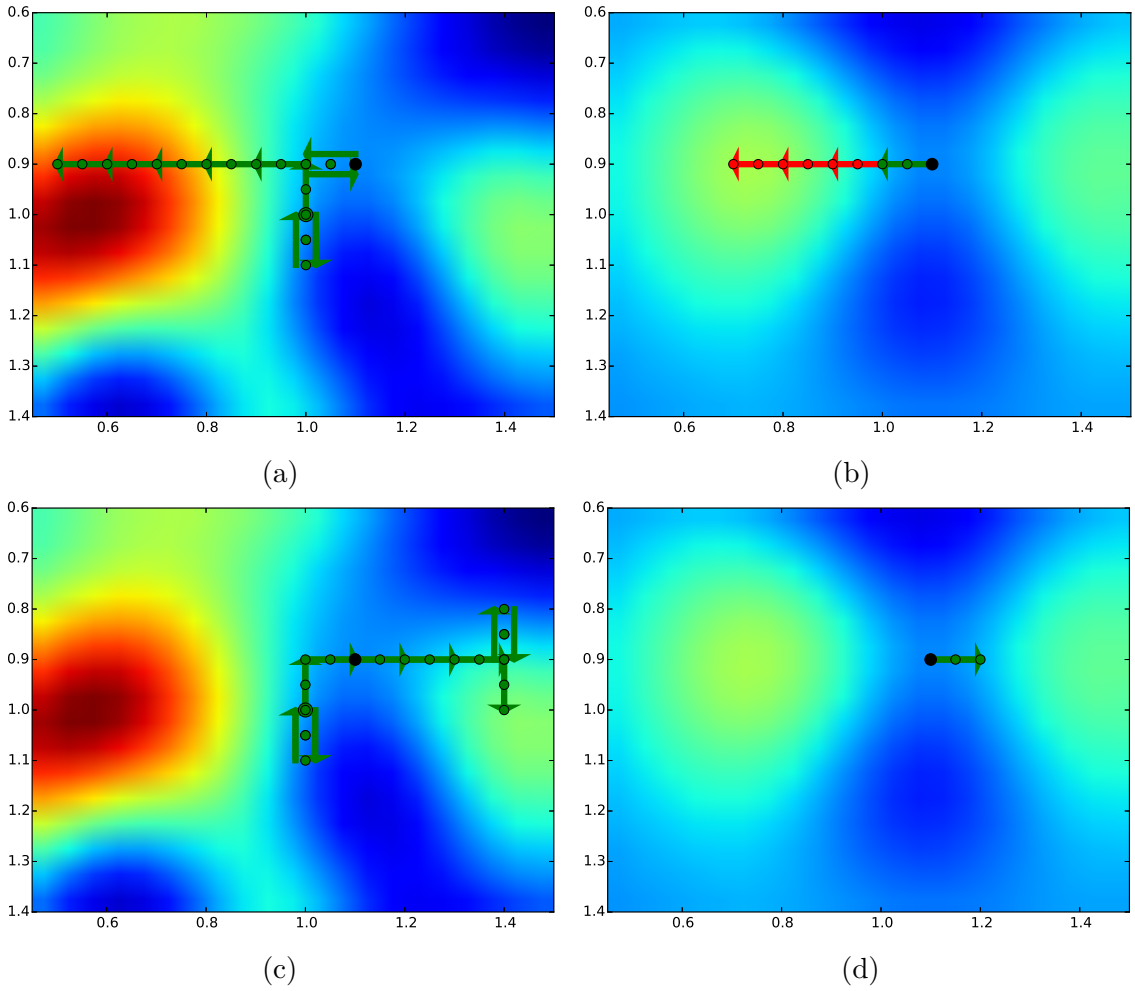


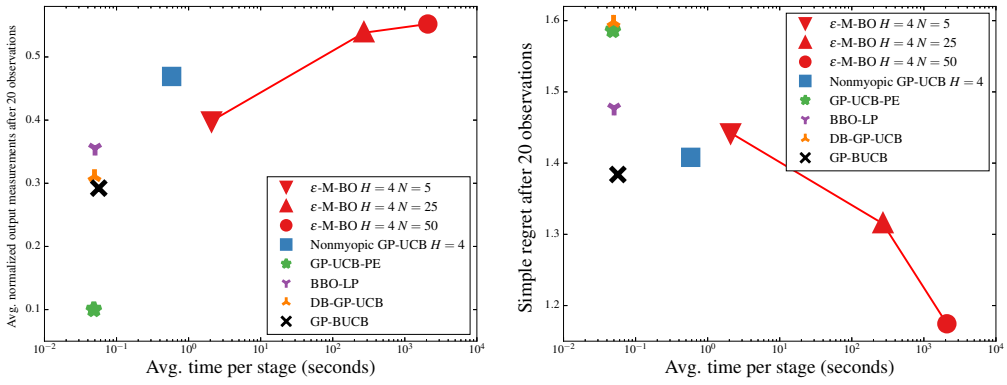
Figure 4.12: Illustrating the behaviors of our nonmyopic ϵ-Macro-BO policy with a lookahead of 8 observations ($H = 4$, $N = 1$) (a,b) vs. greedy/myopic DB-GP-UCB [Daxberger and Low, 2017] (c,d) with macro-action length $\kappa = 2$ in controlling an AUV to gather observations for finding a hotspot (i.e., global maximum) in a simulated plankton density phenomenon.

actions selected by 3 later stages of computing the 4-stage Bellman equations (4.6) at this iteration. That is, ϵ-Macro-BO at iteration $t = 5$ selects the macro-action denoted by the green action, because the trajectory of these 4 macro-actions denoted by arrows results in the highest reward during the planning/computation of Bellman equations due to its direction towards the global maximum. Note that only the macro-action denoted by the green arrow is executed at this iteration. Therefore, the nonmyopic behavior of ϵ-Macro-BO results in turning left to reach the region containing the global maximum. In contrast, DB-GP-UCB selects the macro-action with the highest immediate reward and moves the AUV right towards the local maximum (Fig. 4.12d). So, by utilizing lookahead, our nonmyopic ϵ-Macro-BO policy can outperform the myopic DB-GP-UCB.

4.4.6 Comparison in terms of runtime

In general, nonmyopic methods are expected to be less time-efficient than myopic ones. Fortunately, our nonmyopic ϵ-Macro-BO algorithm with a fixed horizon H offers an advantage of being able to trade off its BO performance for time efficiency by decreasing the number N of stochastic samples. This observation is theoretically validated in Theorem 4.3 and empirically illustrated in Fig. 4.13.

Figs. 4.13a and 4.13b show results of the performances of ϵ-Macro-BO with $H = 4$ (lookahead of 16 observations), $\beta = 0$, and $N = 5, 25, 50$, and the other tested BO algorithms averaged over 35 independent realizations of the simulated plankton density phenomena. It can be observed that as the number of samples N increases, the nonmyopic adaptive BO algorithms outperform the myopic ones. In particular, the performance of ϵ-Macro-BO improves considerably by increasing N: ϵ-Macro-BO with the largest number of samples (i.e., $N = 50$) achieves the largest average normalized output measurements observed by the AUV and smallest simple regret



(a) (b)

Figure 4.13: Graphs of (a) average normalized[12] output measurements observed by AUV, (b) simple regrets achieved by tested BO algorithms vs. average time per iteration for simulated plankton density phenomena.

after 20 observations at the cost of larger average time per iteration. For example, the nonmyopic ϵ-Macro-BO with $H = 4$ and $N = 50$ achieves $0.26\sigma_y$ ($0.083\sigma_y$) more average output measurements and $0.21\sigma_y$ ($0.233\sigma_y$) less simple regret than myopic GP-BUCB (nonmyopic GP-UCB with the same horizon $H = 4$ but assuming most likely observations during planning), but needs 2085.37 (2084.84) more seconds per iteration.

Chapter 5

Black-box adversarial attack automated with BO

This chapter of the thesis proposes and evaluates our novel *Bayesian-Optimization-with-dimension-selection-and-Bayesian-optimal-stopping* (BOS^2) black-box adversarial attack. Firstly, we describe the problem setting of a general black-box adversarial attack in Section 5.1. We then proceed to describing the BOS^2 attack itself (Section 5.2) and start by summarizing it with pseudocode in Section 5.2.1. Specifically, our BOS^2 attack consists of two stages: the dimension BO loop (Section 5.2.2) which selects the dimension of the latent space, and the perturbation BO loop (Section 5.2.3) which searches for the adversarial perturbation in the selected latent space. The key idea of our approach is to increase the attack success rate by using BO for automating both the stages. To boost the query efficiency of our BOS^2 attack, we also use Bayesian optimal stopping [Dai *et al.*, 2019] to early-stop the execution of the perturbation BO loop for those latent dimensions, which will end up under-performing, hence eliminating unnecessary queries to the attacked machine learning model. Finally, the performance of our BOS^2 attack is evaluated in Section 5.3 using MNIST

and CIFAR-10 datasets to show that our method outperforms the existing state-of-the-art black-box adversarial attacks.

5.1 Problem setting

To recall, we consider the black-box setting where the attacker has access only to the outputs of the attacked machine learning model for a given input, but not to the architecture or implementation of the model. Specifically, denote the attacked model as a function $F(x) : \mathbb{R}^D \to [0,1]^C$ where x is a D-dimensional input, which is mapped into one of C classes $\{c_i\}_{i=1,\dots,C}$. Furthermore, we consider a targeted attack, which is more challenging to execute successfully than the untargeted attack (see Section 2.4.2 for the summary of related works on targeted and untargeted attacks). In the setting of targeted black-box attack, for a given benign input $x_0 \in \mathbb{R}^D$ correctly classified by the model F with class c (i.e., $\operatorname{argmax}_{i=1,\dots,C} F(x_0)_{c_i} = c$), we are interested in finding an adversarial example x_{adv}, which is close to x_0, but is classified by the model F with a given target class $c_{target} \neq c$, that is, $\operatorname{argmax}_{i=1,\dots,C} F(x_{adv})_{c_i} = c_{target}$. Here $F(x)_{c_i}$ denotes the score of model F on input x for i-th class c_i.

Closeness of the adversarial input x_{adv} and the benign input x_0 is defined in terms of L_p norm, where p is usually set to either $p = 2$ or $p = \infty$. Following prior state-of-the-art works [Hazan *et al.*, 2017; Alzantot *et al.*, 2019; Ru *et al.*, 2020] we choose $p = \infty$ and hence L_∞ norm to design our attack. Furthermore, we adopt a common practice in the adversarial machine learning literature and instead of searching for the adversarial example x_{adv}, aim to find the adversarial perturbation $\delta \triangleq x_{adv} - x_0$. In this case, the problem in question can be formulated as finding the adversarial perturbation δ satisfying

$$\operatorname*{argmax}_{i=1,\dots,C} F(x_0 + \delta)_{c_i} = c_{target} \quad \text{such that } \delta \in \mathbb{R}^D \text{ and } \|\delta\|_\infty \leq \delta_{max} \tag{5.1}$$

for a pre-defined fixed maximum norm of the adversarial perturbation δ_{max}.

To perform our attack, we re-formulate (5.1) and adopt the following objective function, which was previously used by a number of prior attacks [Alzantot *et al.*, 2019; Ru *et al.*, 2020]. For a given benign input $x_0 \in \mathbb{R}^D$ we aim to maximize the following function:

$$y(\delta) \triangleq \log F(x_0+\delta)_{c_{target}} - \log \sum_{c \neq c_{target}} F(x_0+\delta)_c \quad \text{such that } \delta \in \mathbb{R}^D \text{ and } \|\delta\|_\infty \leq \delta_{max}. \tag{5.2}$$

If $y(\delta) > 0$, then δ is a successful adversarial perturbation satisfying (5.1). To elaborate, in this case, the score $F(x_0+\delta)_{c_{target}}$ of the target class c_{target} is larger than the sum $\sum_{c \neq c_{target}} F(x_0+\delta)_c$ of all other classes' scores due to monotonicity of logarithmic function. As a result, since all the scores $F(x_0+\delta)_c$ are assumed to be in $[0, 1]$ interval, $F(x_0+\delta)_{c_{target}}$ is the highest score among all classes and the input $x_0+\delta$ is classified as c_{target} and, hence, δ is a successful adversarial perturbation. The logarithmic function in (5.2) is used to reduce the numerical instabilities [Carlini and Wagner, 2017; Alzantot *et al.*, 2019; Tu *et al.*, 2019; Ru *et al.*, 2020].

The aim of a black-box adversarial attack is to find a successful adversarial perturbation δ (i.e., such that (5.1) holds). This can be achieved by maximizing $y(\delta)$ (5.2) and finding δ with a positive value of $y(\delta) > 0$, as explained in the paragraph above. The attack is considered successful if such an adversarial perturbation δ is found after using less than T queries to the attacked machine learning model where $T \in \mathbb{N}$ is a given parameter. In the next section we will show how to efficiently search for an adversarial perturbation δ with our BOS^2 attack.

5.2 Bayesian Optimization with dimension selection and Bayesian optimal stopping (BOS^2) attack

BO could be directly applied for performing the black-box attack by maximizing function $y(\delta)$ (5.2). However, the dimension D of the inputs in problem (5.2) is usually too high to effectively apply existing BO algorithms. To make the search for a successful adversarial perturbation δ easier, we perform dimensionality reduction to project the original input space into a latent space of a lower dimension d and search for an adversarial perturbation using BO in this latent space. Formally, denote a function $g : \mathbb{R}^d \to \mathbb{R}^D$ projecting the latent space $\mathbb{R}^d$ back to the original input space $\mathbb{R}^D$. Then (5.2) can be re-written as a lower-dimensional optimization problem in $\mathbb{R}^d$:

$$y(\delta) \triangleq \log F(x_0+g(\delta))_{c_{target}} - \log \sum_{c \neq c_{target}} F(x_0+g(\delta))_c \text{ such that } \delta \in \mathbb{R}^d \text{ and } \|\delta\|_\infty \leq \delta_{max}. \tag{5.3}$$

We use bilinear resizing (bilinear interpolation), which has been shown to be effective by a number of prior works [Alzantot *et al.*, 2019; Tu *et al.*, 2019; Ru *et al.*, 2020], as a dimensionality reduction technique. Other techniques such as autoencoders can be considered instead of bilinear resizing too [Tu *et al.*, 2019], but they require additional resources for training and access to the training dataset of the attacked model. In contrast, bilinear resizing can be performed very fast and requires access only to the current input.

To efficiently maximize the objective function (5.3), we propose a novel *Bayesian-Optimization-with-dimension-selection-and-Bayesian-optimal-stopping* (BOS^2) black-box adversarial attack. In contrast to existing adversarial attacks, we use BO to automate both the selection of the latent space dimension d in (5.3) and the search of

the adversarial perturbation in the selected latent space $\mathbb{R}^d$. To do this, we decompose our BOS^2 attack into two stages. In the first stage we use BO to select the dimension of the latent space d for projecting the high-dimensional input space $\mathbb{R}^D$ into, which we call the *dimension BO loop*. In the second stage (the *perturbation BO loop*), we perform BO with Bayesian optimal stopping in the latent space $\mathbb{R}^d$ in order to find a successful adversarial perturbation δ satisfying $y(\delta) > 0$ (5.3). The two stages of our BOS^2 attack are described in detail in the next two subsections.

5.2.1 BOS^2 attack summary

The pseudocode of our BOS^2 attack is presented in Algorithm 5. As input it receives the total query budget T (i.e., the maximum allowed number of queries to the attacked model), the query budget i_{max} for a fixed latent dimension (i.e., the maximum allowed number of queries to the attacked model for the perturbation BO loop with a fixed latent dimension) and the initial data. Each element of the dataset $\mathcal{D}_{per}$ for the perturbation BO loop is a pair (δ_0, y_0) where δ_0 is a perturbation and y_0 is its corresponding output measurement (5.3). Each element of the dataset $\mathcal{D}_{dim}$ for the dimension BO loop is a pair $\big((d_0, t_0), y_0\big)$ where d_0 is a latent dimension, t_0 is the total number of queries to the attacked model performed at the end of the perturbation BO loop with latent dimension d_0, and y_0 is the best value of output measurement (5.3) found during the perturbation BO loop with latent dimension d_0.

Our BOS^2 attack (Algorithm 5) proceeds as follows: while the current number t of queries to the attacked model is smaller than the total query budget T, it uses BO to select the next latent dimension d^* (the dimension BO loop, line 4). Next it projects the set $\mathcal{D}_{per}$ of previously found perturbations into the latent space of the currently selected dimension d^* and performs BO in this latent space for at most i_{max} iterations (the perturbation BO loop, line 5). The actual number i of BO iterations run by

the perturbation BO loop in line 5 can be smaller than i_{max} since our perturbation BO loop uses Bayesian optimal stopping. After obtaining the set of i perturbations $\{\delta_j\}_{j=1}^{i}$ and their corresponding output measurements $\{y_j\}_{j=1}^{i}$ (5.3), the algorithm augments the new data (lines 6-7). The perturbation BO data $\mathcal{D}_{per}$ is updated with all newly found perturbations and their corresponding output measurements (line 6). In line 7, the dimension BO data $\mathcal{D}_{dim}$ is updated with the current latent dimension d^*, the total number $t + i$ of BO iterations after the execution of line 5 and the best found output measurement $\max\{y_j\}_{j=1}^{i}$ (5.3) found during the execution of the perturbation BO loop in line 5. If the successful adversarial perturbation (i.e., the one satisfying (5.1)) is found at any moment, Algorithm 5 stops immediately and returns this perturbation.

Algorithm 5 BOS^2 attack

1: **Input:** Initial dimension BO data $\mathcal{D}_{dim}$, initial perturbation BO data $\mathcal{D}_{per}$, total query budget T, query budget for a fixed latent dimension i_{max}, list of allowed latent dimensions $d_{allowed}$
2: $t \leftarrow 0$
3: **while** $t < T$ **do**
4: Select a new latent dimension: $d^* \leftarrow \text{DimensionBO}(\mathcal{D}_{dim}, t, i_{max}, d_{allowed})$
5: Perform BO with Bayesian optimal stopping in the latent dimension d^*: $i, \{\delta_j\}_{j=1}^{i}, \{y_j\}_{j=1}^{i} \leftarrow \text{PerturbationBO}(\mathcal{D}_{per}, d^*, i_{max})$
6: Augment perturbation BO data: $\mathcal{D}_{per} \leftarrow \mathcal{D}_{per} \cup \{(\delta_j, y_j)\}_{j=1}^{i}$
7: Augment dimension BO data: $\mathcal{D}_{dim} \leftarrow \mathcal{D}_{dim} \cup \big((d^*, t+i), \max\{y_j\}_{j=1}^{i}\big)$ and update the GP model
8: $t \leftarrow t + i$
9: **return** the successful adversarial perturbation (if found)

5.2.2 The dimension BO loop

In this section we show how the first stage of our BOS^2 attack selects the dimension of the latent space in a principled way (line 4 of Algorithm 5). In contrast to the existing black-box adversarial attacks, which either set this dimension manually or

treat it as a hyperparameter, our BOS^2 attack learns it from the previous data. Specifically, it uses BO to optimize the best output measurement (i.e., the maximum output measurement (5.3)) discovered by the perturbation BO loop as a function of a latent dimension. We choose BO for this optimization problem, because the number of previously explored dimensions for learning is very limited due to the tight total query budget, and BO is known to work well under limited budget constraints. This makes BO a perfect fit for learning the dimension of the latent space for our proposed BOS^2 attack.

Algorithm 6 The dimension BO loop (**DimensionBO** function)

1: **Input:** Dimension BO data $\mathcal{D}_{dim}$, current number of queries to the attacked model t_0, query budget for a fixed latent dimension i_{max}, list of allowed latent dimensions $d_{allowed}$
2: **for** $d_0 \in d_{allowed}$ **do**
3: $\quad x \leftarrow (d_0, t_0 + i_{max})$
4: $\quad$ Compute GP posterior mean and variance $\mu(x, \mathcal{D}_{dim}), \sigma^2(x, \mathcal{D}_{dim})$
5: $\quad$ Compute GP-UCB acquisition function $\alpha(d_0) \leftarrow \mu(x, \mathcal{D}_{dim}) + 2 \cdot \sigma(x, \mathcal{D}_{dim})$
6: $d^* \leftarrow \text{argmax}_{d_0 \in d_{allowed}} \alpha(d_0)$
7: **return** d^*

Algorithm 6 uses two-dimensional tuples (d_0, t_0) where d_0 is a latent dimension and t_0 is the total number of queries to the attacked model at the end of the perturbation BO loop with this latent dimension d_0 (i.e., after executing line 5 of Algorithm 5 with latent dimension d_0) as inputs to BO. The second component t_0 here is required in order to distinguish between the same dimension d_0 being selected multiple times during the execution of our BOS^2 attack. Algorithm 6 uses BO to predict which dimension d^* would produce the best results after executing the perturbation BO loop for i_{max} BO iterations (i.e., after $t_0 + i_{max}$ queries to the attacked model). To do this, it first combines d_0 and $t_0 + i_{max}$ into a tuple for every d_0 in the list of allowed latent dimensions $d_{allowed}$ (line 3). It then computes the GP posterior prediction

(line 4) and GP-UCB[1] acquisition function [Srinivas *et al.*, 2010] (line 5). Finally, it maximizes the GP-UCB acquisition function over all latent dimensions d_0 from the set $d_{allowed}$ (line 6) and returns the maximizing dimension d^*. The dimension BO data $\mathcal{D}_{dim}$ is augmented after the execution of the perturbation BO loop run in the latent space of the selected dimension d^*, and the dimension GP model is updated (line 7 of Algorithm 5).

To illustrate the execution of the BO procedure in Algorithm 6, assume that the current number of queries to the attacked model is $t = 50$ and the query budget for a fixed latent dimension is $i_{max} = 40$. Then line 3 of Algorithm 6 will compute the tuples $(d_0, 90)$ for $d_0 \in d_{allowed}$. Assume that the maximizer in line 6 of Algorithm 6 is $d^* = 196$ and the perturbation BO loop (line 5 of Algorithm 5) runs for $i = 40$ iterations, i.e., no early stopping, and finds the maximum output measurement of -10.3. In this case, in line 7 of Algorithm 5, the data $\mathcal{D}_{dim}$ is augmented with an element $\big((196, 90), -10.3\big)$. Suppose that $d^* = 196$ is selected again at $t = 150$, and the perturbation BO loop (line 5 of Algorithm 5) runs for $i = 23$ iterations, i.e., with early stopping, and discovers the maximum output measurement of -2.4. So, in line 7 of Algorithm 5, the data $\mathcal{D}_{dim}$ is augmented with an element $\big((196, 173), -2.4\big)$ since $t + i = 150 + 23 = 173$.

5.2.3 The perturbation BO loop

After the dimension of the latent space is selected by the dimension BO loop (line 4 of Algorithm 5), our BOS^2 attack proceeds to execute the perturbation BO loop, which is the search for a successful adversarial perturbation in the latent space (Algorithm 7). The perturbation BO loop consists of two major components: BO algorithm and early-stopping. The combination of these two components allows our BOS^2 attack to

[1]We use a constant $\beta_t^{1/2} = 2$ in line 5 of Algorithm 6 as recommended in the source code by Srinivas *et al.* [2010]. For more details about GP-UCB algorithm, see Section 3.2.

search for a successful adversarial perturbation effectively while reducing the number of queries to the attacked model.

The first component of the perturbation BO loop is a BO algorithm performed in the latent space. While the dimension of the latent space is significantly smaller than the dimension of the original input space, it can still be high, making the search with BO challenging. To tackle this challenge, we use Add-GP-UCB algorithm [Kandasamy *et al.*, 2015], which is a popular generalization of GP-UCB [Srinivas *et al.*, 2010] for high-dimensional BO. Add-GP-UCB algorithm approximates the objective function $y(\delta)$ (5.3) by assuming it to be decomposable into a sum of independent local functions $y^{(1)}, \ldots, y^{(M)}$, each of which involves only a small subset of input dimensions:

$$y(\delta) = y^{(1)}(\delta^{(1)}) + \ldots + y^{(M)}(\delta^{(M)})$$

where $\delta^{(j)} \in \mathbb{R}^{d_j}$ are disjoint lower dimensional components of the input vector δ in the latent space $\mathbb{R}^d$ (i.e., $\delta \in \mathbb{R}^d$) such that $\delta^{(1)} \oplus \ldots \oplus \delta^{(M)} = \delta$. Interestingly, by using the additive GP model [Duvenaud *et al.*, 2011], this assumption allows to approximate the GP-UCB acquisition function as a sum of independent local acquisition functions. As a result, the acquisition function used by Add-GP-UCB algorithm at iteration i has the following form:

$$\alpha_i(\delta) \triangleq \sum_{j=1}^{M} \mu_i^{(j)}(\delta^{(j)}) + \beta_i^{1/2} \sigma_i^{(j)}(\delta^{(j)}) \tag{5.4}$$

where each $\mu_i^{(j)}(\delta^{(j)}) + \beta_i^{1/2} \sigma_i^{(j)}(\delta^{(j)})$ is a GP-UCB acquisition function for the respective local function $y^{(j)}(\delta^{(j)})$ at iteration i.

The decomposition (5.4) provides two significant advantages for scalability of BO to higher dimensions. Firstly, the acquisition function $\mu_i^{(j)}(\delta^{(j)}) + \beta_i^{1/2} \sigma_i^{(j)}(\delta^{(j)})$ for each local function $y^{(j)}(\delta^{(j)})$ can be maximized independently, which is much cheaper

than optimizing the standard GP-UCB acquisition function for the full objective function $y(\delta)$ (5.3). Secondly, the BO procedure for each $\delta^{(j)} \in \mathbb{R}^{d_j}$ is less vulnerable to the curse of dimensionality due to the lower dimension of optimization problem in hand. We exploit these advantages for the perturbation BO loop and apply Add-GP-UCB algorithm to search for an adversarial perturbation in the latent space. Since the optimal decomposition of the objective function $y(\delta)$ into $y^{(j)}(\delta^{(j)})$ in (5.4) is unknown, we follow the guideline from the original work of Kandasamy *et al.* [2015] by randomly sampling several decompositions and choosing the one maximizing marginal likelihood.

The second component of the perturbation BO loop of our BOS^2 attack is the Bayesian optimal stopping [Dai *et al.*, 2019], which is used to improve the query efficiency. Specifically, the execution of the perturbation BO loop is early-stopped for those latent dimensions, which will end up under-performing. In contrast, the only adversarial black-box attack which selects the dimension of the latent space adaptively [Ru *et al.*, 2020] updates the latent dimension with a fixed interval of queries to the attacked model. In this case, if the BO loop in the selected latent dimension is performing poorly, the attack of Ru *et al.* [2020] would still keep querying the attacked model unnecessarily till the next update of the latent dimension. As a result, our BOS^2 attack is more query-efficient than the attack of Ru *et al.* [2020], as empirically verified by our experiments in Section 5.3.

Bayesian optimal stopping is a principled mechanism for making a Bayes-optimal decision to stop the execution of an algorithm using a limited number of observations. In each BO iteration i of the perturbation BO loop, the goal of the Bayesian optimal stopping problem is to decide whether to stop and conclude either hypothesis $\theta = \theta_1$ or $\theta = \theta_2$ corresponding to terminal decisions $\mathbb{D}_1$ or $\mathbb{D}_2$, or to gather one more observation via the continuation decision $\mathbb{D}_0$. The decision is made based on minimizing the

expected loss among all decisions

$$\rho_i(\mathbf{y}_{1:i}) \triangleq \min\{\mathbb{E}_{\theta|\mathbf{y}_{1:i}}[l(\mathbb{D}_1,\theta)], \mathbb{E}_{\theta|\mathbf{y}_{1:i}}[l(\mathbb{D}_2,\theta)], c_{\mathbb{D}_0} + \mathbb{E}_{y_{i+1}|\mathbf{y}_{1:i}}[\rho_{i+1}(\mathbf{y}_{1:i+1})]\} \quad (5.5)$$

for $i = 1, \ldots, i_{max}-1$ and $\rho_{i_{max}}(\mathbf{y}_{1:i_{max}}) \triangleq \min\{\mathbb{E}_{\theta|\mathbf{y}_{1:i_{max}}}[l(\mathbb{D}_1,\theta)], \mathbb{E}_{\theta|\mathbf{y}_{1:i_{max}}}[l(\mathbb{D}_2,\theta)]\}$. The loss function l here reflects the cost of making the wrong decision about stopping the execution, $\mathbf{y}_{1:i} \triangleq [y_j]^\top_{1,\ldots,i}$ is the vector of output measurements of the current perturbation BO loop, the first two terms in (5.5) are the expected losses of terminal decisions $\mathbb{D}_1$ and $\mathbb{D}_2$, and the last term is the sum of the immediate cost $c_{\mathbb{D}_0}$ and expected future loss of making the continuation decision $\mathbb{D}_0$ to continue executing the perturbation BO loop.

The problem (5.5) is usually approximately solved using approximate backward induction [Müller *et al.*, 2007]. Its main ideas include using summary statistics to represent the posterior beliefs, discretizing the space of summary statistics, and approximating the expectation terms via sampling [Dai *et al.*, 2019]. After solving the problem (5.5) we obtain a Bayes-optimal decision rule for every iteration of the perturbation BO loop: either early-stop the execution, if the decision rule is one of the terminal decisions $\mathbb{D}_1$ and $\mathbb{D}_2$, or continue the execution for one more iteration (i.e., take the continuation decision $\mathbb{D}_0$).

Let y^* be the maximum output measurement obtained by our BOS^2 attack before the beginning of the current perturbation BO loop. In the context of BO, Bayesian optimal stopping is tasked to decide whether the current run of the perturbation BO loop would result in finding an output measurement better than y^* (i.e., $\max_{j=1,\ldots,i_{max}} y_j > y^*$). Then, the terminal decisions $\mathbb{D}_1$ and $\mathbb{D}_2$ and the continuation decision $\mathbb{D}_0$ are defined as follows [Dai *et al.*, 2019]: $\mathbb{D}_1$ stops and concludes that $\max_{j=1,\ldots,i_{max}} y_j \leq y^*$, $\mathbb{D}_2$ stops and concludes that $\max_{j=1,\ldots,i_{max}} y_j > y^*$ and $\mathbb{D}_1$ continues running the perturbation BO loop for one more iteration. Then, the event

θ mentioned before (5.5) becomes

$$\theta = \begin{cases} \theta_1 & \text{if } \max_{j=1,\dots,i_{max}} y_j \leq y^* \ , \\ \theta_2 & \text{otherwise.} \end{cases}$$

Note that terminal decision $\mathbb{D}_2$ (i.e., the one which stops the execution and concludes that $\max_{j=1,\dots,i_{max}} y_j > y^*$) does not align with the BO objective of sequentially maximizing the objective function. So, when the Bayesian optimal stopping recommends the decision $\mathbb{D}_2$, there is no early stopping and the perturbation BO loop continues for one more iteration.

Algorithm 7 The perturbation BO loop (**PerturbationBO** function)

Input: Perturbation BO data $\mathcal{D}_{per}$, dimension of the latent space d^*, query budget for a fixed latent dimension i_{max}
$\mathcal{D}_{per}^{proj} \leftarrow$ projection of $\mathcal{D}_{per}$ into the latent space $\mathbb{R}^{d^*}$ using bilinear resizing
for $i = 1, \dots, i_0$ **do**
    Find the next perturbation by maximizing Add-GP-UCB acquisition function (5.4): $\delta_i = \operatorname{argmax} \alpha_i(\delta)$
    Compute the output measurement $y_i = y(\delta_i)$ (5.3)
    Augment the data $\mathcal{D}_{per}^{proj} \leftarrow \mathcal{D}_{per}^{proj} \cup (\delta_i, y_i)$
    Update the GP model
Solve Bayesian optimal stopping problem (5.5) to obtain decision rules
$i \leftarrow i_0$
repeat
    Find the next perturbation by maximizing Add-GP-UCB acquisition function (5.4): $\delta_i = \operatorname{argmax} \alpha_i(\delta)$
    Compute the output measurement $y_i = y(\delta_i)$ (5.3)
    Augment the data $\mathcal{D}_{per}^{proj} \leftarrow \mathcal{D}_{per}^{proj} \cup (\delta_i, y_i)$
    Update the GP model
    $i \leftarrow i + 1$
until $i = i_{max}$ or Bayesian optimal stopping decision rule in iteration i outputs the stopping decision $\mathbb{D}_1$
return the number i of iterations run, found perturbations $\{\delta_j\}_{j=1}^{i}$ projected back to the original input space, found output measurements $\{y_j\}_{j=1}^{i}$

The Bayesian optimal stopping method for our BOS^2 attack is based on the BO-BOS algorithm by Dai *et al.* [2019]. BO-BOS algorithm was designed for optimizing the hyperparameters of machine learning models, which require running an iterative training procedure (e.g., stochastic gradient descent for training deep neural networks), with Bayesian early stopping. For our BOS^2 attack, we use the Bayesian optimal stopping procedure of Dai *et al.* [2019] in a novel context. To elaborate, Dai *et al.* [2019] early-stop the training of a machine learning model with a given set of hyperparameters in order to improve the *epoch* efficiency, so their Bayesian optimal stopping method operates on epochs. On the other hand, our BOS^2 attack uses Bayesian optimal stopping to decide on early-stopping the whole BO loop for a fixed dimension of the latent space. That is, it uses Bayesian optimal stopping for improving the *query* efficiency of the method.

To summarize, the perturbation BO loop of our BOS^2 attack (Algorithm 7) proceeds as follows: as input it gets the perturbation BO data $\mathcal{D}_{per}$, the dimension d^* of the latent space selected by the dimension BO loop (line 4 of Algorithm 5) and the query budget i_{max} for a fixed latent dimension (i.e., the maximum allowed number of queries to the attacked model for the perturbation BO loop with the latent dimension d^*). Algorithm 7 first projects the perturbation BO data $\mathcal{D}_{per}$ into the latent space of dimension d^* using bilinear resizing (line 2). After that it runs the BO loop for i_0 initial iterations[2] by optimizing the Add-GP-UCB acquisition function (5.4) without early stopping (lines $3-7$). Next the algorithm solves the Bayesian optimal stopping problem (5.5) using approximate backward induction [Dai *et al.*, 2019] to obtain the decision rules (line 8). After obtaining the decision rules the algorithm continues running the BO loop (lines $11-15$) either until the query budget i_{max} is exhausted (i.e., no early-stopping) or until the stopping decision $\mathbb{D}_1$ is discovered (line 16).

[2]For our experiments i_0 is set as $i_0 \triangleq i_{max}/5$.

5.3 Experimental results

In this section, we empirically evaluate the performance of our BOS^2 attack using two state-of-the-art datasets for image classification MNIST [LeCun *et al.*, 1998] and CIFAR-10 [Krizhevsky *et al.*, 2009].

The performances of our algorithm are compared with those of the state-of-the-art black-box adversarial attacks such as GenAttack [Alzantot *et al.*, 2019], ZOO [Chen *et al.*, 2017], AutoZOOM [Tu *et al.*, 2019] and BayesOpt [Ru *et al.*, 2020]. We use open source implementations of these algorithms with the default parameter settings provided by the authors.

We use four performance metrics. The first and the most important performance metric is the attack success rate (ASR), i.e., the percentage of the runs where the attack discovered a successful adversarial example. The other three metrics are related to the query count (that is, the number of queries to the attacked model): maximum, mean and median.

We limit the maximum allowed number of queries to the attacked model to $T = 900$ queries for our experiments. The list of allowed latent space dimensions in Algorithm 6 is set to $d_{allowed} = \{6 \times 6 \times c, 8 \times 8 \times c, 10 \times 10 \times c, 12 \times 12 \times c, 14 \times 14 \times c, 16 \times 16 \times c, 18 \times 18 \times c\}$ where c is the number of channels ($c = 1$ for MNIST and $c = 3$ for CIFAR-10) to ensure fair comparison with BayesOpt [Ru *et al.*, 2020]. These dimensions are used to apply bilinear resizing on the original inputs.

For the dimension BO loop, GP with the Matérn kernel ($\nu = 5/2$) is used. The GP hyperparameters are learned using maximum likelihood estimation [Rasmussen and Williams, 2006] and updated after every iteration. The acquisition function used for BO procedure is GP-UCB [Srinivas *et al.*, 2010] with a constant parameter $\beta_t^{1/2} = 2$.

For the perturbation BO loop, the query budget for a fixed latent dimension i_{max} is set to $i_{max} = 40$. The number i_0 of initial iterations (line 3 of Algorithm 7) is set to

$i_0 = 8$. The BO loop is performed using an additive GP [Duvenaud *et al.*, 2011] and Add-GP-UCB algorithm [Kandasamy *et al.*, 2015] with $M = 12$ components (5.4). The optimal decomposition in (5.4) is learned by maximizing marginal likelihood over 20 randomly selected decompositions. A new decomposition is learned every time the dimension of the latent space is updated. The GP hyperparameters are learned using maximum likelihood estimation [Rasmussen and Williams, 2006] and updated every 5 iterations. The parameters used for the Bayesian optimal stopping procedure are those recommended by the open source implementation of BO-BOS [Dai *et al.*, 2019].

To initialize our BOS^2 attack, we first select 3 latent dimensions $d_{init} = \{6 \times 6 \times c, 14 \times 14 \times c, 18 \times 18 \times c\}$ where c is the number of channels, then randomly sample 10 perturbations in each of these 3 latent spaces and obtain the corresponding output measurements (5.3). The dimension BO loop is then initialized with 3 tuples $\big((d, 10), y\big)$ where $d \in d_{init}$ and y is maximum output measurement from all initial perturbations sampled in the latent space of dimension d. The perturbation BO loop is initialized with all 30 perturbations and their corresponding output measurements. Note that the work of Ru *et al.* [2020] also uses 30 inputs to initialize their attack.

Both the attacked models used in our experiments were originally proposed by Carlini and Wagner [2017] and used by a number of black-box attacks after that [Chen *et al.*, 2017; Tu *et al.*, 2019; Alzantot *et al.*, 2019; Ru *et al.*, 2020]. Specifically, the models are pre-trained CNN image classifiers with test accuracy of 99.5% for MNIST dataset and test accuracy of 80% for CIFAR-10 dataset.

To perform our experiments, we randomly select a number of images from test data: 50 images for MNIST dataset and 20 images for CIFAR-10 dataset. We select only those images, which are correctly classified by the attacked model. We then perform targeted attacks on these images. Each image in our experiments is attacked 9 times, targeting all classes except the true class. This setting results in 450 attack instances for MNIST dataset and 180 attack instances for CIFAR-10 dataset.

The maximum norm of the adversarial perturbation δ_{max} is set as $\delta_{max} = 0.3$ for MNIST and $\delta_{max} = 0.05$ for CIFAR-10 to ensure fair comparison with the previous works [Alzantot *et al.*, 2019; Ru *et al.*, 2020] which used the same values of δ_{max}.

5.3.1 MNIST dataset

Table 5.1 shows the performances of tested black-box attacks with the maximum allowed number of queries to the attacked model $T = 900$ and $\delta_{max} = 0.3$ on MNIST dataset. It can be observed that our BOS^2 attack achieves the highest attack success rate (ASR) of all the tested attacks. Specifically, the ASR of our attack is significantly higher than the ASR of all other attacks, except BayesOpt [Ru *et al.*, 2020]. Comparing to BayesOpt [Ru *et al.*, 2020], our BOS^2 attack not only achieves higher ASR, but is also more query efficient: the median query count (mean query count) of BOS^2 attack is 20.25% (7.04%) smaller than that of BayesOpt [Ru *et al.*, 2020].

Table 5.1: Performances of the tested black-box attacks with the maximum allowed number of queries to the attacked model $T = 900$ queries and $\delta_{max} = 0.3$ on MNIST dataset. The results are averaged over 450 attack instances. For ZOO and AutoZOOM, max count, mean count and median count values refer to the initially found successful adversarial perturbation.

Attack method	ASR	Max count	Mean count	Median count
BOS^2	99%	897	108.75	63
GenAttack	64%	881	428.66	396
ZOO	1%	95	65.88	67
AutoZOOM	1%	77	72.5	72.5
BayesOpt	98%	817	116.99	79

The results of GenAttack [Alzantot *et al.*, 2019] are consistent with those reported in the original work: GenAttack achieves the ASR 100% on MNIST with a median query of 996 queries (Table 1 in Alzantot *et al.* [2019]), which is larger than the maximum allowed number of queries $T = 900$ we use in our experimental setting.

ZOO [Chen *et al.*, 2017] and AutoZOOM [Tu *et al.*, 2019] both achieve a very low ASR of 1%. This can be explained by the procedure of these attacks: they first find an initial successful adversarial perturbation with a large distortion and then use the subsequent queries to the attacked model to refine the initial perturbation. However, the number of queries required to achieve the distortion similar to $\delta_{max} = 0.3$ is very large (i.e., in the scale of thousands of queries), which is much higher than the maximum allowed number of queries $T = 900$ we use in our experimental setting. We consider the attacks produced by ZOO [Chen *et al.*, 2017] and AutoZOOM [Tu *et al.*, 2019] successful only if their final adversarial perturbation discovered is within $\delta_{max} = 0.3$ to the original image according to L_∞ norm.

Table 5.2: Performances of the tested black-box attacks with the maximum allowed number of queries to the attacked model $T = 900$ queries and $\delta_{max} = 0.3$ on MNIST dataset. The results are averaged over 450 attack instances.

Attack method	ASR	Max count	Mean count	Median count
BOS^2	99%	897	108.75	63
BOS^2 (no stopping)	99%	895	133.34	88
BayesOpt	98%	817	116.99	79

We also empirically investigate the impact of the selection of the latent space dimension using BO (Section 5.2.2) and Bayesian optimal stopping (Section 5.2.3) on the performance of our BOS^2 attack. To do this, we compare 3 attack methods: BOS^2 attack, BOS^2 attack run without the use of Bayesian optimal stopping and BayesOpt [Ru *et al.*, 2020], which is similar to the latter one, but optimizes the dimension of the latent space as a hyperparameter. It can be observed from Table 5.2 that BOS^2 attack run without the use of Bayesian optimal stopping achieves higher ASR than BayesOpt [Ru *et al.*, 2020], which verifies our claim that selecting the dimension of the latent space using BO can increase the ASR. Furthermore, it can be observed from Table 5.2 that BOS^2 attack run without the use of Bayesian optimal stopping results in 39.68% larger median query count (22.61% larger mean query

count) than BOS^2 attack, which shows that Bayesian optimal stopping is able to boost the query efficiency.

5.3.2 CIFAR-10 dataset

Table 5.3 presents the performances of tested black-box attacks with the maximum allowed number of queries to the attacked model $T = 900$ queries and $\delta_{max} = 0.05$ on CIFAR-10 dataset. The results are consistent with those for MNIST dataset in the previous section: Our BOS^2 attack achieves the highest ASR of all the tested attacks. Furthermore, our BOS^2 attack outperforms BayesOpt [Ru *et al.*, 2020] by achieving higher ASR, 7% smaller median query count and 9.54% smaller mean query count.

Table 5.3: Performances of the tested black-box attacks with the maximum allowed number of queries to the attacked model $T = 900$ queries and $\delta_{max} = 0.05$ on CIFAR-10 dataset. The results are averaged over 180 attack instances. For ZOO and AutoZOOM, max count, mean count and median count values refer to the initially found successful adversarial perturbation.

Attack method	ASR	Max count	Mean count	Median count
BOS^2	84%	877	253.5	164.5
GenAttack	64%	891	410.7	386
ZOO	20%	452	142.28	104
AutoZOOM	1%	176	169.5	169.5
BayesOpt	79%	883	247.49	176.5

The results of GenAttack [Alzantot *et al.*, 2019] agree with the original paper since, according to Alzantot *et al.* [2019], GenAttack requires a median query count of 804 in order to achieve the ASR of 96.5% on CIFAR-10 (Table 1 in Alzantot *et al.* [2019]). The small values of ASR for ZOO [Chen *et al.*, 2017] and AutoZOOM [Tu *et al.*, 2019] are explained by the larger number of queries required by these attacks, as mentioned in the previous section.

We also examine the effect of the selection of the latent space dimension using BO and Bayesian optimal stopping on the performance of our BOS^2 attack. Similarly to

the experiments on MNIST dataset in the previous section, we compare BOS^2 attack, BOS^2 attack run without the use of Bayesian optimal stopping and BayesOpt [Ru *et al.*, 2020]. The observations from the results in Table 5.4 are consistent with those for MNIST dataset in Table 5.2. Specifically, BOS^2 attack run without Bayesian optimal stopping outperforms BayesOpt [Ru *et al.*, 2020] in terms of ASR, which again shows that selecting the dimension of the latent space using BO can increase the ASR. It can be also observed from Table 5.4 that BOS^2 attack run without Bayesian optimal stopping requires 10% larger median query count and almost similar mean query count compared to those of BOS^2 attack, which again supports our claim that Bayesian optimal stopping improves the query efficiency of our BOS^2 attack.

Table 5.4: Performances of the tested black-box attacks with the maximum allowed number of queries to the attacked model $T = 900$ queries and $\delta_{max} = 0.05$ on CIFAR-10 dataset. The results are averaged over 180 attack instances.

Attack method	ASR	Max count	Mean count	Median count
BOS^2	84%	877	253.5	164.5
BOS^2 (no stopping)	83%	896	248.03	181
BayesOpt	79%	883	247.49	176.5

Chapter 6

Conclusion

This thesis has investigated the following question:

How can BO be scaled up to satisfy the additional requirements of new real-world applications?

6.1 Summary of contributions

While working towards a satisfactory answer to the question stated above, we have been able to make the following progress:

- *We proposed PO-GP-UCB, which is the first algorithm for BO in the outsourced setting with differential privacy and theoretical performance guarantee [Kharkovskii et al., 2020a].*
- *We presented a principled multi-staged Bayesian sequential decision algorithm for nonmyopic adaptive BO for hotspot sampling in spatially varying phenomena that exploits macro-actions for scaling up to a further lookahead comparing to existing BO algorithms [Kharkovskii et al., 2020b].*

- *We designed a novel algorithm for performing a black-box adversarial attack that uses BO for automating both the selection of the latent space dimension and the search of the adversarial perturbation in the selected latent space in order to increase the attack success rate.*

All of the items above are substantiated by the following specific contributions:

6.1.1 Private Outsourced BO (Chapter 3)

- *Performance guarantee.* We established a theoretical upper bound on the regret similar to that of the original GP-UCB algorithm [Srinivas *et al.*, 2010].

- *Privacy-preserving property.* We formally proved the privacy-preserving property of our algorithm using the celebrated differential privacy framework and empirically demonstrated the ability of our algorithm to achieve state-of-the-art privacy guarantees in the single-digit range.

- *Analysis of privacy-utility trade-off.* We analyzed how our theoretical results are amenable to interpretations regarding the privacy-utility trade-off by tuning different parameters of our PO-GP-UCB algorithm.

- *Empirical evaluation.* We used both synthetic and real-world datasets to show the empirical effectiveness of our algorithm.

6.1.2 Nonmyopic BO with Macro-Actions (Chapter 4)

- *Novel acquisition function.* We generalized GP-UCB to a novel acquisition function defined with respect to a nonmyopic adaptive macro-action policy.

- *Novel nonmyopic adaptive BO algorithm with performance guarantee.* Since our proposed acquisition function is intractable to be optimized exactly due to an

uncountable set of candidate outputs, we proposed a nonmyopic adaptive ϵ-*Bayes-optimal macro-action BO* (ϵ-Macro-BO) algorithm for hotspot sampling in spatially varying phenomena by approximating the acquisition function using stochastic sampling. We showed that our algorithm can achieve any arbitrary user-specified loss bound ϵ, which requires only a polynomial number of samples in the length of macro-actions in each planning stage.

- *Anytime algorithm.* To perform nonmyopic adaptive BO in real time, we proposed an asymptotically optimal anytime variant of our ϵ-Macro-BO algorithm with a performance guarantee.

- *Empirical evaluation.* Our experiments with synthetic and real-world datasets revealed that a relatively small sample size (N=100-300) is needed for ϵ-Macro-BO and its anytime variant to outperform state-of-the-art BO algorithms.

6.1.3 Adversarial attack automated with BO (Chapter 5)

- *BO for increasing the attack success rate.* To increase the attack success rate, we used BO for automating both the latent space dimension using GP-UCB algorithm [Srinivas *et al.*, 2010] and the search of adversarial perturbation in the selected latent space using Add-GP-UCB algorithm [Kandasamy *et al.*, 2015].

- *Bayesian optimal stopping for improving the query efficiency.* We used Bayesian optimal stopping [Dai *et al.*, 2019] to boost the query efficiency of our BOS^2 attack. Specifically, we early-stopped the execution of Add-GP-UCB algorithm [Kandasamy *et al.*, 2015] in those latent spaces, which would end up under-performing, hence eliminating unnecessary queries to the attacked machine learning model.

- *Empirical evaluation.* We used the famous MNIST and CIFAR-10 datasets

to demonstrate that our proposed BOS^2 algorithm outperforms the existing algorithms for black-box adversarial attacks.

6.2 Future Work

This section proposes and discusses potential research directions that could be pursued as continuation of the works described in this thesis.

6.2.1 Private Outsourced BO (Chapter 3)

A natural way to extend this work would be investigating whether PO-GP-UCB can be extended for privately releasing the output measurements y_t. To this end, the work of Hall *et al.* [2013] which provides a way for DP release of functional data could potentially be applied.

Another potential direction would be to further improve the privacy guarantee of our algorithm. For example, it would be interesting to research whether the work of Kenthapadi *et al.* [2013] on DP random projection can be used as a privacy-preserving mechanism in our outsourced BO framework.

6.2.2 Nonmyopic BO with Macro-Actions (Chapter 4)

There are a few directions that can be pursued as continuation of this work. One of them would be to consider the macro-actions of variable length, similarly to options in reinforcement learning [Barto and Mahadevan, 2003; Konidaris and Barto, 2007; Stolle and Precup, 2002]. It would be interesting to investigate whether such a generalization of our ϵ-Macro-BO algorithm would still be amenable to theoretical analysis. Furthermore, it is also worth considering whether the empirical BO performance of ϵ-Macro-BO and its anytime variant can be improved by using macro-action generation

algorithms [He *et al.*, 2011] instead of random selection, as we did in our experiments.

The GP hyperparameters are learned a priori using maximum likelihood estimation [Rasmussen and Williams, 2006] for all our experiments (except simulated plankton density), which is a common practice in the nonmyopic BO literature. However, such an approach might result in performance loss due to model overfitting/misspecification. To this end, another potential direction for extending this work would be to mitigate these negative effects by considering Bayesian treatment of GP hyperparameters with stochastic sampling, similarly to the work of Hoang *et al.* [2014].

6.2.3 Adversarial attack automated with BO (Chapter 5)

It would be interesting to find out whether the two-stage procedure we used in our BOS^2 attack could be transformed into a general BO algorithm on images. Such an algorithm, for instance, could use BO to select parameters for transforming the given set of images first (e.g., parameters used for image augmentation techniques such as rotation, color space transformations or resizing) and then perform BO on the transformed images. It could be also worth investigating whether our BOS^2 attack is amenable to theoretical performance analysis.

Another potential direction could be exploring if performance of our BOS^2 attack can be improved by using other high-dimensional BO algorithms [Hoang *et al.*, 2018; Rolland *et al.*, 2018] or other methods for early-stopping [Domhan *et al.*, 2015; Klein *et al.*, 2017].

Bibliography

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proc. of Conf. on Computer and Communications Security*, pages 308–318, 2016.

[Alzantot *et al.*, 2019] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani B Srivastava. Genattack: Practical black-box attacks with gradient-free optimization. In *Proc. of the Genetic and Evolutionary Computation Conference*, pages 1111–1119, 2019.

[Azimi *et al.*, 2010] Javad Azimi, Alan Fern, and Xiaoli Z Fern. Batch Bayesian optimization via simulation matching. In *Proc. NIPS*, pages 109–117, 2010.

[Balcan *et al.*, 2006] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning*, 65(1):79–94, 2006.

[Baraniuk *et al.*, 2008] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[Barto and Mahadevan, 2003] Andrew G Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(4):341–379, 2003.

[Bergstra *et al.*, 2011] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Proc. NeurIPS*, pages 2546–2554, 2011.

[Blocki *et al.*, 2012] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The Johnson-Lindenstrauss transform itself preserves differential privacy. In *Proc. IEEE Annual Symposium on Foundations of Computer Science*, pages 410–419, 2012.

[Bogunovic *et al.*, 2016] Ilija Bogunovic, Jonathan Scarlett, and Volkan Cevher. Time-varying Gaussian process bandit optimization. In *Proc. AISTATS*, pages 314–323, 2016.

[Bojarski *et al.*, 2016] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[Boucheron *et al.*, 2013] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence.* Oxford University Press, 2013.

[Brendel *et al.*, 2017] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[Brochu *et al.*, 2010] Eric Brochu, Vlad M Cora, and Nando de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[Bull, 2011] Adam D Bull. Convergence rates of efficient global optimization algorithms. *JMLR*, 12:2879–2904, 2011.

[Burkard and Lagesse, 2017] Cody Burkard and Brent Lagesse. Analysis of causative attacks against SVMs learning from data streams. In *Proc. ACM on International Workshop on Security And Privacy Analytics*, pages 31–36, 2017.

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE symposium on security and privacy*, pages 39–57, 2017.

[Chandrasekaran *et al.*, 2012] Venkat Chandrasekaran, Benjamin Recht, Pablo A Parrilo, and Alan S Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[Chen *et al.*, 2015] Jie Chen, Kian Hsiang Low, Yujian Yao, and Patrick Jaillet. Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems. *IEEE T-ASE*, 12(3):901–921, 2015.

[Chen *et al.*, 2017] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017.

[Chen *et al.*, 2018] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Proc. AAAI*, 2018.

[Chevalier and Ginsbourger, 2013] Clément Chevalier and David Ginsbourger. Fast computation of the multi-points expected improvement with applications in batch selection. In *Proc. 7th International Conference on Learning and Intelligent Optimization*, pages 59–69, 2013.

[Choi *et al.*, 2012] Joon-Ho Choi, Vivian Loftness, and Azizan Aziz. Post-occupancy evaluation of 20 office buildings as basis for future IEQ standards and guidelines. *Energy and Buildings*, 46:167–175, 2012.

[Chong *et al.*, 2005] Miao Chong, Ajith Abraham, and Marcin Paprzycki. Traffic accident analysis using machine learning paradigms. *Informatica (Slovenia)*, 29(1):89–98, 2005.

[Contal *et al.*, 2013] Emile Contal, David Buffoni, Alexandre Robicquet, and Nicolas Vayatis. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Proc. ECML/PKDD*, pages 225–240, 2013.

[Cover and Thomas, 2006] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, 2nd edition, 2006.

[Dai *et al.*, 2019] Zhongxiang Dai, Haibin Yu, Bryan Kian Hsiang Low, and Patrick Jaillet. Bayesian optimization meets bayesian optimal stopping. In *Proc. ICML*, pages 1496–1506, 2019.

[Dallaire *et al.*, 2009] Patrick Dallaire, Camille Besse, Stephane Ross, and Brahim Chaib-draa. Bayesian reinforcement learning in continuous POMDPs with Gaussian processes. In *Proc. IEEE/RSJ IROS*, pages 2604–2609, 2009.

[Daxberger and Low, 2017] Erik A Daxberger and Bryan Kian Hsiang Low. Distributed batch Gaussian process optimization. In *Proc. ICML*, pages 951–960, 2017.

[Desautels *et al.*, 2014] Thomas Desautels, Andreas Krause, and Joel W Burdick. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *JMLR*, 15:4053–4103, 2014.

[Dewancker *et al.*, 2016] Ian Dewancker, Michael McCourt, Scott Clark, Patrick Hayes, Alexandra Johnson, and George Ke. Evaluation system for a Bayesian optimization service. arxiv:1605.06170, 2016.

[Domhan *et al.*, 2015] Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Proc. IJCAI*, 2015.

[Duvenaud *et al.*, 2011] David K Duvenaud, Hannes Nickisch, and Carl E Rasmussen. Additive Gaussian processes. In *Proc. NIPS*, pages 226–234, 2011.

[Dwork and Roth, 2014] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[Dwork *et al.*, 2006] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. Theory of Cryptography*, pages 265–284, 2006.

[Eykholt *et al.*, 2018] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proc. CVPR*, pages 1625–1634, 2018.

[Feng *et al.*, 2019] Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: Generating training time adversarial data with auto-encoder. In *Proc. NeurIPS*, pages 11994–12004, 2019.

[Foulds *et al.*, 2016] James Foulds, Joseph Geumlek, Max Welling, and Kamalika Chaudhuri. On the theory and practice of privacy-preserving Bayesian data analysis. In *Proc. UAI*, pages 192–201, 2016.

[Garnett *et al.*, 2010] Roman Garnett, Michael A Osborne, and Stephen J Roberts. Bayesian optimization for sensor set selection. In *Proc. ACM/IEEE international conference on information processing in sensor networks*, pages 209–219, 2010.

[Golub and Van Loan, 1996] G. H. Golub and C.-F. Van Loan. *Matrix Computations*. Johns Hopkins Univ. Press, 3rd edition, 1996.

[González *et al.*, 2016a] Javier González, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. Batch Bayesian optimization via local penalization. In *Proc. AISTATS*, pages 648–657, 2016.

[González *et al.*, 2016b] Javier González, Michael Osborne, and Neil D Lawrence. GLASSES: Relieving the myopia of Bayesian optimisation. In *Proc. AISTATS*, pages 790–799, 2016.

[Goodfellow *et al.*, 2014] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[Gu and Rigazio, 2014] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

[Hall *et al.*, 2013] Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Differential privacy for functions and functional data. *JMLR*, 14(1):703–727, 2013.

[Hardt and Roth, 2012] Moritz Hardt and Aaron Roth. Beating randomized response on incoherent matrices. In *Proc. Symposium on Theory of Computing*, pages 255–1268, 2012.

[Hazan *et al.*, 2017] Tamir Hazan, George Papandreou, and Daniel Tarlow. Adversarial perturbations of deep neural networks. 2017.

[He *et al.*, 2010] Ruijie He, Emma Brunskill, and Nicholas Roy. PUMA: Planning under uncertainty with macro-actions. In *Proc. AAAI*, pages 1089–1095, 2010.

[He *et al.*, 2011] Ruijie He, Emma Brunskill, and Nicholas Roy. Efficient planning under uncertainty with macro-actions. *JAIR*, 40:523–570, 2011.

[Heaton *et al.*, 2017] James B Heaton, Nick G Polson, and Jan Hendrik Witte. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1):3–12, 2017.

[Hennig and Schuler, 2012] Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *JMLR*, 13:1809–1837, 2012.

[Hernández-Lobato *et al.*, 2014] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Proc. NIPS*, pages 918–926, 2014.

[Hoang *et al.*, 2014] Trong Nghia Hoang, Bryan Kian Hsiang Low, Patrick Jaillet, and Mohan Kankanhalli. Nonmyopic ϵ-Bayes-Optimal Active Learning of Gaussian Processes. In *Proc. ICML*, pages 739–747, 2014.

[Hoang *et al.*, 2018] Trong Nghia Hoang, Quang Minh Hoang, Ruofei Ouyang, and Kian Hsiang Low. Decentralized high-dimensional Bayesian optimization with factor graphs. In *Proc. AAAI*, 2018.

[Hoffman *et al.*, 2014] Matthew Hoffman, Bobak Shahriari, and Nando de Freitas. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *Proc. AISTATS*, pages 365–374, 2014.

[Ilyas *et al.*, 2018] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *Proc. ICML*, 2018.

[Jagannathan *et al.*, 2012] Geetha Jagannathan, Krishnan Pillaipakkamnatt, and Rebecca N Wright. A practical differentially private random decision tree classifier. *Trans. Data Privacy*, 5(1):273–295, 2012.

[Johnson and Lindenstrauss, 1984] William B Johnson and Joram Lindenstrauss. Extensions of Lipschitz maps into a Hilbert space. *Contemporary Mathematics*, 26(2):189–206, 1984.

[Kandasamy *et al.*, 2015] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *Proc. ICML*, pages 295–304, 2015.

[Kenthapadi *et al.*, 2013] Krishnaram Kenthapadi, Aleksandra Korolova, Ilya Mironov, and Nina Mishra. Privacy via the johnson-lindenstrauss transform. *Journal of Privacy and Confidentiality*, 5(1):39–71, 2013.

[Kharkovskii *et al.*, 2020a] Dmitrii Kharkovskii, Zhongxiang Dai, and Kian Hsiang Low. Private outsourced Bayesian optimization. In *Proc. ICML*, 2020.

[Kharkovskii *et al.*, 2020b] Dmitrii Kharkovskii, Chun Kai Ling, and Kian Hsiang Low. Nonmyopic Gaussian process optimization with macro-actions. In *Proc. AISTATS*, 2020.

[Klein *et al.*, 2017] Aaron Klein, Stefan Falkner, Jost Tobias Springenberg, and Frank Hutter. Learning curve prediction with Bayesian neural networks. In *Proc. ICLR*, 2017.

[Koh and Liang, 2017] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proc. ICML*, pages 1885–1894, 2017.

[Konidaris and Barto, 2007] George Konidaris and Andrew G Barto. Building portable options: Skill transfer in reinforcement learning. In *Proc. IJCAI*, pages 895–900, 2007.

[Krause and Ong, 2011] Andreas Krause and Cheng S Ong. Contextual Gaussian process bandit optimization. In *Proc. NIPS*, pages 2447–2455, 2011.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Kurakin *et al.*, 2016] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[Kusner *et al.*, 2015] Matt Kusner, Jacob Gardner, Roman Garnett, and Kilian Weinberger. Differentially private Bayesian optimization. In *Proc. ICML*, pages 918–927, 2015.

[Kwon *et al.*, 2018] Hyun Kwon, Yongchul Kim, Hyunsoo Yoon, and Daeseon Choi. Random untargeted adversarial example on deep neural network. *Symmetry*, 10(12):738, 2018.

[Lam and Willcox, 2017] Remi Lam and Karen Willcox. Lookahead Bayesian optimization with inequality constraints. In *Proc. NIPS*, 2017.

[Lam *et al.*, 2016] Remi Lam, Karen Willcox, and David H Wolpert. Bayesian optimization with a finite budget: An approximate dynamic programming approach. In *Proc. NIPS*, 2016.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Leonard *et al.*, 2007] Naomi Ehrich Leonard, Derek A Paley, Francois Lekien, Rodolphe Sepulchre, David M Fratantoni, and Russ E Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.

[Li *et al.*, 2017] Cheng Li, David Rubín de Celis Leal, Santu Rana, Sunil Gupta, Alessandra Sutti, Stewart Greenhill, Teo Slezak, Murray Height, and Svetha Venkatesh. Rapid Bayesian optimisation for synthesis of short polymer fiber materials. *Scientific reports*, 7(1):5683, 2017.

[Lim *et al.*, 2011] Zhan Lim, Lee Sun, and David Hsu. Monte Carlo value iteration with macro-actions. In *Proc. NIPS*, pages 1287–1295, 2011.

[Ling *et al.*, 2016] Chun Kai Ling, Kian Hsiang Low, and Patrick Jaillet. Gaussian process planning with Lipschitz continuous reward functions: Towards unifying Bayesian optimization, active learning, and beyond. In *Proc. AAAI*, pages 1860–1866, 2016.

[Linial *et al.*, 1995] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[Liu *et al.*, 2016] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[Liu *et al.*, 2018] Yi Liu, Jie Ling, Zhusong Liu, Jian Shen, and Chongzhi Gao. Finger vein secure biometric template generation based on deep learning. *Soft Computing*, 22(7):2257–2265, 2018.

[Lizotte *et al.*, 2007] Daniel J Lizotte, Tao Wang, Michael H Bowling, and Dale Schuurmans. Automatic gait optimization with Gaussian process regression. In *Proc. IJCAI*, pages 944–949, 2007.

[Marchant *et al.*, 2014] Roman Marchant, Fabio Ramos, and Scott Sanner. Sequential Bayesian optimisation for spatial-temporal monitoring. In *Proc. UAI*, pages 553–562, 2014.

[Martinez-Cantin *et al.*, 2007] Ruben Martinez-Cantin, Nando de Freitas, Arnaud Doucet, and José A Castellanos. Active policy learning for robot planning and exploration under uncertainty. In *Proc. RSS*, pages 321–328, 2007.

[Mei and Zhu, 2015] Shike Mei and Xiaojin Zhu. Using machine teaching to identify optimal training-set attacks on machine learners. In *Proc. AAAI*, 2015.

[Moosavi-Dezfooli *et al.*, 2016] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proc. CVPR*, pages 2574–2582, 2016.

[Mori *et al.*, 2005] N. Mori, M. Takeda, and K. Matsumoto. A comparison study between genetic algorithms and bayesian optimize algorithms by novel indices. In *Proc. of Conference on Genetic and evolutionary computation*, pages 1485–1492, 2005.

[Müller *et al.*, 2007] Peter Müller, Don A Berry, Andy P Grieve, Michael Smith, and Michael Krams. Simulation-based sequential Bayesian design. *Journal of statistical planning and inference*, 137(10):3140–3150, 2007.

[Newell *et al.*, 2014] Andrew Newell, Rahul Potharaju, Luojie Xiang, and Cristina Nita-Rotaru. On the practicality of integrity attacks on document-level sentiment analysis. In *Proc. Workshop on Artificial Intelligent and Security*, pages 83–93, 2014.

[Ngai *et al.*, 2011] Eric WT Ngai, Yong Hu, Yiu Hing Wong, Yijun Chen, and Xin Sun. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision support systems*, 50(3):559–569, 2011.

[Nguyen *et al.*, 2018] Thanh Nguyen, Sunil Gupta, Santu Rana, and Svetha Venkatesh. A privacy preserving Bayesian Optimization with high efficiency. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 543–555, 2018.

[Osborne *et al.*, 2009] Michael A Osborne, Roman Garnett, and Stephen J Roberts. Gaussian processes for global optimization. In *Proc. 3rd International Conference on Learning and Intelligent Optimization*, 2009.

[Papadimitriou *et al.*, 1995] Christos H Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, 61(2):217–235, 1995.

[Papernot *et al.*, 2016] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[Papernot *et al.*, 2017a] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *Proc. ICLR*, 2017.

[Papernot *et al.*, 2017b] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proc. ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[Pennington *et al.*, 2016] J Timothy Pennington, Marguerite Blum, and FP Chavez. Seawater sampling by an autonomous underwater vehicle: "Gulper" sample validation for nitrate, chlorophyll, phytoplankton, and primary production. *Limnol. Oceanogr.: Methods*, 14(1):14–23, 2016.

[Petersen and Pedersen, 2012] K. B. Petersen and M. S. Pedersen. *The Matrix Cookbook*. 2012.

[Poupart *et al.*, 2006] Pascal Poupart, Nikos Vlassis, Jesse Hoey, and Kevin Regan. An analytic solution to discrete Bayesian reinforcement learning. In *Proc. ICML*, pages 697–704, 2006.

[Quiñonero-Candela and Rasmussen, 2005] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

[Rasmussen and Williams, 2006] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.

[Rolland *et al.*, 2018] Paul Rolland, Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. High-dimensional Bayesian optimization via additive models with overlapping groups. *Proc. AISTATS*, 2018.

[Ross *et al.*, 2008] Stephane Ross, Brahim Chaib-draa, and Joelle Pineau. Bayesian reinforcement learning in continuous POMDPs with application to robot navigation. In *Proc. IEEE ICRA*, pages 2845–2851, 2008.

[Ru *et al.*, 2020] Binxin Ru, Adam Cobb, Arno Blaas, and Yarin Gal. Bayesopt adversarial attack. In *Proc. ICLR*, 2020.

[Rubinstein *et al.*, 2012] Benjamin IP Rubinstein, Peter L Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for SVM learning. *Journal of Privacy and Confidentiality*, 4(1):65–100, 2012.

[Sarwate and Chaudhuri, 2013] Anand D Sarwate and Kamalika Chaudhuri. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE Signal Processing Magazine*, 30(5):86–94, 2013.

[Sessa *et al.*, 2019] Pier Giuseppe Sessa, Ilija Bogunovic, Maryam Kamgarpour, and Andreas Krause. No-regret learning in unknown games with correlated payoffs. In *Proc. NeurIPS*, 2019.

[Shah and Ghahramani, 2015] Amar Shah and Zoubin Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Proc. NIPS*, pages 3312–3320, 2015.

[Shahriari *et al.*, 2016] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[Smith *et al.*, 2018] Michael Smith, Mauricio Álvarez, Max Zwiessele, and Neil D Lawrence. Differentially private regression with Gaussian Processes. In *Proc. AISTATS*, pages 1195–1203, 2018.

[Snoek *et al.*, 2012] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Proc. NeurIPS*, pages 2951–2959, 2012.

[Srinivas *et al.*, 2010] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. ICML*, pages 1015–1022, 2010.

[Stewart and Sun, 1990] G. W. Stewart and J. Sun. *Matrix Perturbation Theory.* Academic Press, 1990.

[Stolle and Precup, 2002] Martin Stolle and Doina Precup. Learning options in reinforcement learning. In *Proc. International Symposium on Abstraction, Reformulation, and Approximation*, pages 212–223, 2002.

[Suya *et al.*, 2017] Fnu Suya, Yuan Tian, David Evans, and Paolo Papotti. Query-limited black-box attacks to classifiers. In *NeurIPS workshop*, 2017.

[Swersky *et al.*, 2013] Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task Bayesian optimization. In *Proc. NeurIPS*, pages 2004–2012, 2013.

[Taboga, 2017] M. Taboga. *Lectures on probability theory and mathematical statistics.* CreateSpace Independent Publishing Platform, 2017. http://www.statlect.com.

[Thornton *et al.*, 2013] Chris Thornton, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proc. SIGKDD*, pages 847–855, 2013.

[Tu *et al.*, 2019] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. AutoZOOM: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proc. AAAI*, pages 742–749, 2019.

[Vazquez and Bect, 2010] Emmanuel Vazquez and Julien Bect. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *J. Statistical Planning and Inference*, 140(11):3088–3095, 2010.

[Villemonteix *et al.*, 2009] Julien Villemonteix, Emmanuel Vazquez, and Eric Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *J. Glob. Optim.*, 44(4):509–534, 2009.

[Wu and Frazier, 2016] Jian Wu and Peter Frazier. The parallel knowledge gradient method for batch Bayesian optimization. In *Proc. NIPS*, pages 3126–3134, 2016.

[Wu *et al.*, 2019] Aming Wu, Yahong Han, Quanxin Zhang, and Xiaohui Kuang. Untargeted adversarial attack via expanding the semantic gap. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 514–519, 2019.

[Xiao *et al.*, 2012] Han Xiao, Huang Xiao, and Claudia Eckert. Adversarial label flips attack on support vector machines. In *ECAI*, pages 870–875, 2012.

[Xiao *et al.*, 2015] Huang Xiao, Battista Biggio, Blaine Nelson, Han Xiao, Claudia Eckert, and Fabio Roli. Support vector machines under adversarial label contamination. *Neurocomputing*, 160:53–62, 2015.

[Yu *et al.*, 2013] Shipeng Yu, Faisal Farooq, Alexander Van Esbroeck, Glenn Fung, Vikram Anand, and Balaji Krishnapuram. Predicting readmission risk with institution specific prediction models. In *Proc. IEEE Int. Conf. on Healthcare Informatics*, pages 415–420, 2013.

[Zhao *et al.*, 2019] Pu Zhao, Sijia Liu, Pin-Yu Chen, Nghia Hoang, Kaidi Xu, Bhavya Kailkhura, and Xue Lin. On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method. In *Proc. ICCV*, pages 121–130, 2019.

Appendix A

Appendix for Chapter 3

A.1 Proof of Lemma 3.1

Fix $x, x' \in \mathcal{X}$. It follows from Theorem 3.5 by setting vector $y = (x - x')^\top$ and $r \times d$ matrix $M' = M^\top$ that

$$\begin{aligned} & 1 - 2\exp(-\nu^2 r/8) \\ & \leq P\Big((1-\nu)\|(x-x')^\top\|^2 \leq r^{-1}\|M^\top(x-x')^\top\|^2 \leq (1+\nu)\|(x-x')^\top\|^2\Big) \\ & = P\Big((1-\nu)\|x-x'\|^2 \leq r^{-1}\|xM - x'M\|^2 \leq (1+\nu)\|x-x'\|^2\Big). \end{aligned} \tag{A.1}$$

Since there are no more than $n^2/2$ pairs of inputs $x, x' \in \mathcal{X}$, applying the union bound to (A.1) gives that the probability of

$$(1-\nu)\|x-x'\|^2 \leq r^{-1}\|xM - x'M\|^2 \leq (1+\nu)\|x-x'\|^2$$

for all $x, x' \in \mathcal{X}$ is at least $1 - n^2\exp(-\nu^2 r/8)$.

To guarantee that the probability of $(1-\nu)\|x-x'\|^2 \leq r^{-1}\|Mx - Mx'\|^2 \leq (1+\nu)\|x-x'\|^2$ for all $x, x' \in \mathcal{X}$ is at least $1-\mu$, the value of r has to satisfy the following inequality:

$$1 - n^2\exp(-\nu^2 r/8) \geq 1 - \mu,$$

which is equivalent to $r \geq 8\log(n^2/\mu)/\nu^2$.

A.2 Privacy guarantee of Algorithm 2

A.2.1 Comparison between Algorithm 2 and Algorithm 3 of Blocki et al. [2012]

There are several important differences between our Algorithm 2 and the work of Blocki *et al.* [2012]. Firstly, Algorithm 3 of Blocki *et al.* [2012] outputs a DP estimate $r^{-1}\tilde{\mathcal{X}}^\top M^\top M\tilde{\mathcal{X}}$ (in the notations of Algorithm 2) of the covariance matrix $r^{-1}\mathcal{X}^\top\mathcal{X}$, while our Algorithm 2 outputs a DP transformation $r^{-1/2}\mathcal{X}M$ (or $r^{-1/2}\tilde{\mathcal{X}}M$) of the original dataset $\mathcal{X}$. However, the authors of Blocki *et al.* [2012] prove the privacy guarantee (see Theorem 4.1, p. 13 of their paper) by showing that releasing $\tilde{\mathcal{X}}^\top M^\top$ (using matrix M of size $r \times n$) preserves DP and then apply the post-processing property of DP to reconstruct $r^{-1}\tilde{\mathcal{X}}^\top M^\top M\tilde{\mathcal{X}}$. This observation allows us to modify their proof for our Algorithm 2. Additionally, matrix $\tilde{\mathcal{X}}^\top M^\top$ (in the notations of Algorithm 2) in the proof of Blocki *et al.* [2012] has size $d \times r$, while matrices $r^{-1/2}\mathcal{X}M$ and $r^{-1/2}\tilde{\mathcal{X}}M$ returned by our Algorithm 2 have size $n \times r$, which requires us to modify the proof of Blocki *et al.* [2012]. These modifications are discussed in Section A.2.2 below.

Secondly, Algorithm 3 of Blocki *et al.* [2012] does not have the "if/else" condition (line 6 of Algorithm 2) and always increases the singular values as in line 9 of Algorithm 2, since the authors are able to offset the bias introduced to the estimate of covariance of the dataset along a given dimension by increasing the singular values. Specifically, they do it by subtracting ω^2 from the computed estimate (see Algorithm 4 in Blocki *et al.* [2012]). For our case, however, the distances between the original inputs from the dataset $\mathcal{X}$ are no longer approximately the same as the distances between their images from the dataset $\mathcal{Z}$ when $\sigma_{min}(\mathcal{X}) < \omega$ (i.e., the "else" clause, line 8 of Algorithm 2), as shown in Theorem 3.7. Therefore, the case of $\sigma_{min}(\mathcal{X}) < \omega$ results in a slightly different regret bound (see Theorem 3.8 and Remark 3.2) and requires us to introduce the "if/else" condition into Algorithm 2. Introducing such an "if/else" condition, however, does not affect the proof of Theorem 4.1 of Blocki *et al.* [2012] and our proof: the "if" clause (line 6 of Algorithm 2) is stated in the Corollary (see p. 17 of Blocki *et al.* [2012]), while the "else" clause (line 8 of Algorithm 2) is proved in Theorem 4.1 of Blocki *et al.* [2012].

A.2.2 Proof of Theorem 3.6

Fix two neighboring datasets $\mathcal{X}$ and $\mathcal{X}'$. Let $E \triangleq \mathcal{X}' - \mathcal{X}$, such that E is a rank 1 matrix. Without loss of generality, we assume that in the definition of neighboring datasets (Definition 3.4) $\|x_{(i^*)} - x'_{(i^*)}\| = 1$. Then we can write E as the outer product $E = e_{i^*}v^\top$ where e_{i^*} is the indicator vector of row i^* and v is the vector of norm 1.

Then the singular values of E are exactly $\{1, 0, \ldots, 0\}$ (see Blocki *et al.* [2012], p. 14).

Similar to Theorem 4.1 of Blocki *et al.* [2012], the proof is composed of two stages. For the first stage we work under the premise that both and $\mathcal{X}$ and $\mathcal{X}'$ have singular values no less than ω (the "if" clause, line 6 of Algorithm 2). For the second stage we denote $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{X}}'$ as the respective matrices from "else" clause (line 8 of Algorithm 2) and show what adaptations are needed to make the proof follow through.

We prove the theorem for the scaled output of the "if" clause of Algorithm 2 $\mathcal{X}M$ (the post-processing property of DP can be applied after that to reconstruct $r^{-1/2}\mathcal{X}M$). $\mathcal{X}M$ is composed of r columns each is an i.i.d. sample from $\mathcal{X}Y$ where $Y \sim \mathcal{N}(0, I_{d\times d})$. The following lemma is similar to Claim 4.3 of Blocki *et al.* [2012](p. 14):

Lemma A.1. *Let $\epsilon > 0$, $\delta \in (0,1)$, $r \in \mathbb{N}$, $d \in \mathbb{N}$, two neighboring datasets $\mathcal{X}$ and $\mathcal{X}'$ and Y sampled from $\mathcal{N}(0, I_{d\times d})$ be given. Fix $\epsilon_0 \triangleq \epsilon/\sqrt{4r\log(2/\delta)}$ and $\delta_0 \triangleq \delta/(2r)$. Denote*

$$S \triangleq \{\xi \in \mathbb{R}^n : \exp(-\epsilon_0) PDF_{\mathcal{X}'Y}(\xi) \leq PDF_{\mathcal{X}Y}(\xi) \leq \exp(\epsilon_0) PDF_{\mathcal{X}'Y}(\xi)\}$$

where PDF is the probability density function. Then $P(S) \geq 1 - \delta_0$.

Proof. Similar to the proof of Claim 4.3 of Blocki *et al.* [2012], first we formally define the PDF of the two distributions. We apply the fact that $\mathcal{X}Y$ and $\mathcal{X}'Y$ are linear transformations of $\mathcal{N}(0, I_{d\times d})$.

$$\mathrm{PDF}_{\mathcal{X}Y}(\xi) = \frac{1}{\sqrt{(2\pi)^n \det(\mathcal{X}\mathcal{X}^\top)}} \exp\Big(-\frac{1}{2}\xi^\top(\mathcal{X}\mathcal{X}^\top)^{-1}\xi\Big)$$
$$\mathrm{PDF}_{\mathcal{X}'Y}(\xi) = \frac{1}{\sqrt{(2\pi)^n \det(\mathcal{X}'\mathcal{X}'^\top)}} \exp\Big(-\frac{1}{2}\xi^\top(\mathcal{X}'\mathcal{X}'^\top)^{-1}\xi\Big).$$

If the matrix $\mathcal{X}\mathcal{X}^\top$ (all the reasoning here is exactly the same for $\mathcal{X}'\mathcal{X}'^\top$) is not full-rank, the SVD allows us to use similar notation to denote the generalizations of the inverse and of the determinant: The Moore-Penrose inverse of any square matrix M is $M^\dagger \triangleq V\Sigma^{-1}U^\top$ where $M = U\Sigma V^\top$ is the SVD of matrix M, and the pseudo-determinant of M is $\widetilde{det}(M) \triangleq \Pi_{i=1}^{rank(M)}\sigma_i(M)$ where $\sigma_i(M)$ are the singular values of matrix M. Furthermore, if $\mathcal{X}\mathcal{X}^\top$ has non-trivial kernel space (i.e., is not invertible) then $\mathrm{PDF}_{\mathcal{X}Y}$ in the equation above is technically undefined. However, if we restrict ourselves only to the subspace $\mathcal{V} = (\mathrm{Ker}(\mathcal{X}\mathcal{X}^\top))^\perp$, then $\mathrm{PDF}^{\mathcal{V}}_{\mathcal{X}Y}$ is defined over $\mathcal{V}$ and $\mathrm{PDF}^{\mathcal{V}}_{\mathcal{X}Y}(\xi) \triangleq \frac{1}{\sqrt{(2\pi)^{rank(\mathcal{X}\mathcal{X}^\top)}\widetilde{\det}(\mathcal{X}\mathcal{X}^\top)}} \exp\Big(-\frac{1}{2}\xi^\top(\mathcal{X}\mathcal{X}^\top)^\dagger\xi\Big)$
From now on, we omit the superscript from the PDF and refer to the above function as the PDF of $\mathcal{X}Y$. See p. 4–5 of Blocki *et al.* [2012] for more details.

Similar to the proof of Claim 4.3 of Blocki *et al.* [2012], first we show that

$$\exp(-\epsilon_0/2) \leq \sqrt{\frac{\det(\mathcal{X}'\mathcal{X}'^\top)}{\det(\mathcal{X}\mathcal{X}^\top)}} \leq \exp(\epsilon_0/2).$$

The proof copies the derivation of eq. 4 in Blocki *et al.* [2012] (p. 15) with replacing A to $\mathcal{X}^\top$, A' to $\mathcal{X}'^\top$, x to ξ and swapping n and d where necessary.

Next we prove an analogue of eq. 5 of Claim 4.3 of Blocki *et al.* [2012]:

$$P_\xi\left(\frac{1}{2}|\xi^\top\big((\mathcal{X}\mathcal{X}^\top)^{-1} - (\mathcal{X}'\mathcal{X}'^\top)^{-1}\big)\xi| \geq \epsilon_0/2\right) \leq \delta_0. \tag{A.2}$$

To do this:

$$\begin{aligned}
&\xi^\top\big((\mathcal{X}\mathcal{X}^\top)^{-1} - (\mathcal{X}'\mathcal{X}'^\top)^{-1}\big)\xi \\
&= \xi^\top\big((\mathcal{X}\mathcal{X}^\top)^{-1} - (\mathcal{X}'\mathcal{X}'^\top)^{-1}\mathcal{X}\mathcal{X}^\top(\mathcal{X}\mathcal{X}^\top)^{-1}\big)\xi \\
&= \xi^\top\big((\mathcal{X}\mathcal{X}^\top)^{-1} - (\mathcal{X}'\mathcal{X}'^\top)^{-1}(\mathcal{X}' - E)(\mathcal{X}' - E)^\top(\mathcal{X}\mathcal{X}^\top)^{-1}\big)\xi \\
&= \xi^\top\big((\mathcal{X}\mathcal{X}^\top)^{-1} - (\mathcal{X}'\mathcal{X}'^\top)^{-1}(\mathcal{X}'\mathcal{X}'^\top - E\mathcal{X}'^\top - \mathcal{X}'E^\top + EE^\top)(\mathcal{X}\mathcal{X}^\top)^{-1}\big)\xi \\
&= \xi^\top\big((\mathcal{X}\mathcal{X}^\top)^{-1} - (\mathcal{X}\mathcal{X}^\top)^{-1} - (\mathcal{X}'\mathcal{X}'^\top)^{-1}(-E\mathcal{X}'^\top - \mathcal{X}'E^\top + EE^\top)(\mathcal{X}\mathcal{X}^\top)^{-1}\big)\xi \\
&= \xi^\top(\mathcal{X}'\mathcal{X}'^\top)^{-1}(E\mathcal{X}'^\top + \mathcal{X}'E^\top - EE^\top)(\mathcal{X}\mathcal{X}^\top)^{-1}\xi \\
&= \xi^\top(\mathcal{X}'\mathcal{X}'^\top)^{-1}(E\mathcal{X}^\top + \mathcal{X}'E^\top)(\mathcal{X}\mathcal{X}^\top)^{-1}\xi
\end{aligned} \tag{A.3}$$

where the second and the last equalities are due to $E = \mathcal{X}' - \mathcal{X}$. The expression in the last line of (A.3) is very similar to the one in the derivation of eq. 5 in Blocki *et al.* [2012] (p. 15). The difference is that in order for the proof to go through, we need to multiply $(\mathcal{X}'\mathcal{X}'^\top)^{-1}$ by $\mathcal{X}\mathcal{X}^\top(\mathcal{X}\mathcal{X}^\top)^{-1}$ in the second line of (A.3), while the original proof of Blocki *et al.* [2012] multiplies $(\mathcal{X}^\top\mathcal{X})^{-1}$ by $\mathcal{X}'^\top\mathcal{X}'(\mathcal{X}'^\top\mathcal{X}')^{-1}$ (in our notations), see eq. in the bottom of p. 15 of Blocki *et al.* [2012].

Now denoting singular value decompositions of $\mathcal{X} = U\Sigma V^\top$ and $\mathcal{X}' = U'\Lambda V'^\top$, and the fact that $E = e_{i^*}v^\top$, we continue (A.3):

$$\begin{aligned}
&\xi^\top(\mathcal{X}'\mathcal{X}'^\top)^{-1}(E\mathcal{X}^\top + \mathcal{X}'E^\top)(\mathcal{X}\mathcal{X}^\top)^{-1}\xi \\
&= \xi^\top(\mathcal{X}'\mathcal{X}'^\top)^{-1}E\mathcal{X}^\top(\mathcal{X}\mathcal{X}^\top)^{-1}\xi + \xi^\top(\mathcal{X}'\mathcal{X}'^\top)^{-1}\mathcal{X}'E^\top(\mathcal{X}\mathcal{X}^\top)^{-1}\xi \\
&= \xi^\top(U'\Lambda V'^\top V'\Lambda U'^\top)^{-1}(e_{i^*}\cdot v^\top V\Sigma U^\top)(U\Sigma V^\top V\Sigma U^\top)^{-1}\xi \\
&+\xi^\top(U'\Lambda V'^\top V'\Lambda U'^\top)^{-1}(U'\Lambda V'^\top v\cdot e_{i^*}^\top)(U\Sigma V^\top V\Sigma U^\top)^{-1}\xi \\
&= \xi^\top U'\Lambda^{-2}U'^\top e_{i^*}\cdot v^\top V\Sigma^{-1}U^\top\xi + \xi^\top U'\Lambda^{-1}V'^\top v\cdot e_{i^*}^\top U\Sigma^{-2}U^\top\xi
\end{aligned} \tag{A.4}$$

where the last equality is due to the properties of singular value decomposition.

So now, assume ξ is sampled from $\mathcal{X}'Y$ (the case of $\mathcal{X}Y$ is symmetric). That is, assume that we've sampled χ from $Y \sim \mathcal{N}(0, I_{d\times d})$ and we have $\xi = \mathcal{X}'\chi = U'\Lambda V'^\top\chi$

and equivalently $\xi = (\mathcal{X} + E)\chi = U\Sigma V^\top \chi + e_{i^*} v^\top \chi$. Plugging it into (A.4) gives:

$$\begin{aligned}
&|\xi^\top U' \Lambda^{-2} U'^\top e_{i^*} \cdot v^\top V \Sigma^{-1} U^\top \xi + \xi^\top U' \Lambda^{-1} V'^\top v \cdot e_{i^*}^\top U \Sigma^{-2} U^\top \xi| \\
&= |(U' \Lambda V'^\top \chi)^\top U' \Lambda^{-2} U'^\top e_{i^*} \cdot v^\top V \Sigma^{-1} U^\top (U \Sigma V^\top \chi + e_{i^*} v^\top \chi) \\
&+ (U' \Lambda V'^\top \chi)^\top U' \Lambda^{-1} V'^\top v \cdot e_{i^*}^\top U \Sigma^{-2} U^\top (U \Sigma V^\top \chi + e_{i^*} v^\top \chi)| \\
&= |\chi^\top V' \Lambda U'^\top U' \Lambda^{-2} U'^\top e_{i^*} \cdot v^\top V \Sigma^{-1} U^\top (U \Sigma V^\top \chi + e_{i^*} v^\top \chi) \\
&+ \chi^\top V' \Lambda U'^\top U' \Lambda^{-1} V'^\top v \cdot e_{i^*}^\top U \Sigma^{-2} U^\top (U \Sigma V^\top \chi + e_{i^*} v^\top \chi)| \\
&\leq term_1 \cdot term_2 + term_3 \cdot term_4
\end{aligned}$$

where for $i = 1, 2, 3, 4$ we have $term_i = |vec_i \cdot \chi|$ and

$$\begin{aligned}
&vec_1 \\
&= (V' \Lambda U'^\top U' \Lambda^{-2} U'^\top e_{i^*})^\top \\
&= (V' \Lambda^{-1} U'^\top e_{i^*})^\top
\end{aligned}$$

so $\|vec_1\| \leq 1/\lambda_d$;

$$\begin{aligned}
&vec_2 \\
&= v^\top V \Sigma^{-1} U^\top (U \Sigma V^\top + e_{i^*} v^\top) \\
&= v^\top + v^\top V \Sigma^{-1} U^\top e_{i^*} v^\top
\end{aligned}$$

so $\|vec_2\| \leq 1 + 1/\sigma_d$;

$$\begin{aligned}
&vec_3 \\
&= (V' \Lambda U'^\top U' \Lambda^{-1} V'^\top v)^\top \\
&= v^\top
\end{aligned}$$

so $\|vec_3\| \leq 1$;

$$\begin{aligned}
&vec_4 \\
&= e_{i^*}^\top U \Sigma^{-2} U^\top (U \Sigma V^\top + e_{i^*} v^\top) \\
&= e_{i^*}^\top U \Sigma^{-1} V^\top + e_{i^*}^\top U \Sigma^{-2} U^\top e_{i^*} v^\top
\end{aligned}$$

so $\|vec_4\| \leq 1/\sigma_d + 1/\sigma_d^2$ where σ_d and λ_d are the smallest singular values of $\mathcal{X}$ and $\mathcal{X}'$, respectively. The remainder of the proof now follows the proof of Claim 4.3 of Blocki *et al.* [2012] with replacing A to $\mathcal{X}^\top$, A' to $\mathcal{X}'^\top$, x to ξ and swapping n and d where necessary. □

For the second stage we assume that "else" clause (line 8 of Algorithm 2) is applied and denote $\tilde{\mathcal{X}} \triangleq U\sqrt{\Sigma^2 + \omega^2 I_{n \times d}} V^\top$ and $\tilde{\mathcal{X}}' \triangleq U'\sqrt{\Lambda^2 + \omega^2 I_{n \times d}} V'^\top$. The theorem requires an analogue of Lemma A.1 to hold, which depends on the following two

conditions:

$$\exp(-\epsilon_0/2) \leq \sqrt{\frac{\det(\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top)}{\det(\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)}} \leq \exp(\epsilon_0/2). \tag{A.5}$$

$$P_\xi\left(\frac{1}{2}|\xi^\top\big((\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1} - (\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top)^{-1}\big)\xi| \geq \epsilon_0/2\right) \leq \delta_0. \tag{A.6}$$

Derivation of (A.5) copies the derivation of eq. 6 in Blocki *et al.* [2012] (p. 16). To derive (A.6), we start with an observation regarding $\mathcal{X}'\mathcal{X}'^\top$ and $\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top$:

$$\begin{aligned}
&\mathcal{X}'\mathcal{X}'^\top = (\mathcal{X}+E)(\mathcal{X}+E)^\top = \mathcal{X}\mathcal{X}^\top + \mathcal{X}'E^\top + E\mathcal{X}^\top \\
&\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top = U(\Sigma^2+\omega^2 I)U^\top = U\Sigma^2U^\top + \omega^2 I = \mathcal{X}\mathcal{X}^\top + \omega^2 I \\
&\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top = U'(\Lambda^2+\omega^2 I)U'^\top = U'\Lambda^2U'^\top + \omega^2 I = \mathcal{X}'\mathcal{X}'^\top + \omega^2 I \\
&\implies \tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top - \tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top = \mathcal{X}'E^\top + E\mathcal{X}^\top.
\end{aligned} \tag{A.7}$$

Now we can follow the same outline as in the proof of (A.2). Fix ξ, then

$$\begin{aligned}
&\xi^\top\big((\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1} - (\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top)^{-1}\big)\xi \\
&= \xi^\top\big((\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1} - (\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top)^{-1}\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top(\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1}\big)\xi \\
&= \xi^\top\big((\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1} - (\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top)^{-1}(\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top - \mathcal{X}'E^\top - E\mathcal{X}^\top)(\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1}\big)\xi \\
&= \xi^\top\big((\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1} - (\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1} - (\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top)^{-1}(-\mathcal{X}'E^\top - E\mathcal{X}^\top)(\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1}\big)\xi \\
&= \xi^\top(\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top)^{-1}(\mathcal{X}'E^\top + E\mathcal{X}^\top)(\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1}\xi \\
&= \xi^\top(\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top)^{-1}(\mathcal{X}'E^\top - EE^\top + EE^\top + E\mathcal{X}^\top)(\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1}\xi \\
&= \xi^\top(\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top)^{-1}((\mathcal{X}'-E)E^\top + E(\mathcal{X}^\top + E^\top))(\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1}\xi \\
&= \xi^\top(\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top)^{-1}(\mathcal{X}'-E)v\cdot e_{i^*}^\top(\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1}\xi \\
&+\xi^\top(\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top)^{-1}e_{i^*}\cdot v^\top(\mathcal{X}^\top + E^\top)(\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1}\xi
\end{aligned} \tag{A.8}$$

where the second equality follows from (A.7) and the last equality follows from $E = e_{i^*}v^\top$. The expression in the last line of (A.8) is very similar to the one in the derivation of equation in Blocki *et al.* [2012] (p. 17, second equation array from the top). The difference is that in order for the proof to go trhough, we need to multiply $(\tilde{\mathcal{X}}'\tilde{\mathcal{X}}'^\top)^{-1}$ by $\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top(\tilde{\mathcal{X}}\tilde{\mathcal{X}}^\top)^{-1}$ in the second line of (A.8), while the original proof of Blocki *et al.* [2012] multiplies $(\tilde{\mathcal{X}}^\top\tilde{\mathcal{X}})^{-1}$ by $\tilde{\mathcal{X}}'^\top\tilde{\mathcal{X}}'(\tilde{\mathcal{X}}'^\top\tilde{\mathcal{X}}')^{-1}$ (in our notations), see second equation array from the top, p. 17 of Blocki *et al.* [2012]. The remainder of the proof now follows the proof of Theorem 4.1 of Blocki *et al.* [2012] (p. 17).

A.3 Proof of Theorem 3.7

Proof. Fix $x, x' \in \mathcal{X}$ and their images $z, z' \in \mathcal{Z}$. If $\sigma_{min}(\mathcal{X}) \geq \omega$, according to Algorithm 2, $\mathcal{Z} = r^{-1/2}\mathcal{X}M$ (line 7) and

$$\begin{aligned}&\|z - z'\|^2\\ &= \|r^{-1/2}xM - r^{-1/2}x'M\|^2\\ &= r^{-1}\|xM - x'M\|^2\end{aligned}$$

and Lemma 3.1 can be immediately applied.

If $\sigma_{min}(\mathcal{X}) < \omega$, according to Algorithm 2, $\mathcal{Z} = r^{-1/2}\tilde{\mathcal{X}}M$ (line 10) and

$$\begin{aligned}&\|z - z'\|^2\\ &= \|r^{-1/2}\tilde{x}M - r^{-1/2}\tilde{x}'M\|^2\\ &= r^{-1}\|\tilde{x}M - \tilde{x}'M\|^2\\ &\leq (1+\nu)\|\tilde{x} - \tilde{x}'\|^2\\ &\leq (1+\nu)(1+\omega^2/\sigma_{min}^2(\mathcal{X}))\|x - x'\|^2\end{aligned}$$

where the first inequality follows from Lemma 3.1 and the second inequality follows from Lemma A.6. Similarly,

$$\begin{aligned}&\|z - z'\|^2\\ &= \|r^{-1/2}\tilde{x}M - r^{-1/2}\tilde{x}'M\|^2\\ &= r^{-1}\|\tilde{x}M - \tilde{x}'M\|^2\\ &\geq (1-\nu)\|\tilde{x} - \tilde{x}'\|^2\\ &\geq (1-\nu)\|x - x'\|^2\end{aligned}$$

where the first inequality follows from Lemma 3.1 and the second inequality follows from Lemma A.6. □

A.4 Bounding the covariance change

Theorem A.1. *Let a dataset $\mathcal{X} \subset \mathbb{R}^d$ be given and $\sigma_{min}(\mathcal{X}) > 0$ be the smallest singular value of $\mathcal{X}$. Let $r \in \mathbb{N}$ be the input parameter of Algorithm 2, a dataset $\mathcal{Z} \subset \mathbb{R}^r$ be the output of Algorithm 2 and ω be defined in line 5 of Algorithm 2. Let $\mathrm{d} = diam(\mathcal{X})/l$ where $diam(\mathcal{X})$ is the diameter of the dataset $\mathcal{X}$. Let $\nu \in (0, 1/2)$, $\mu \in (0, 1)$ be given. If $\nu \leq 2/\mathrm{d}^2$ and $r \geq 8\log(n^2/\mu)/\nu^2$, then the probability of*

$$|k_{zz'} - k_{xx'}| \leq C \cdot k_{xx'}$$

for all $x, x' \in \mathcal{X}$ and their images under Algorithm 2 $z, z' \in \mathcal{Z}$ is at least $1-\mu$ where

$$C \triangleq \begin{cases} \nu \mathrm{d}^2 & \text{if } \sigma_{min}(\mathcal{X}) \geq \omega, \\ \max\left(\nu\mathrm{d}^2, 1 - \exp\left(-0.5(\nu + \nu\omega^2/\sigma^2_{min}(\mathcal{X}) + \omega^2/\sigma^2_{min}(\mathcal{X}))\mathrm{d}^2\right)\right) & \text{otherwise.} \end{cases} \tag{A.9}$$

Remark A.1. It immediately follows from Theorem A.1 that the probability of $k_{zz'} \leq (1+C) \cdot k_{xx'}$ for all $x, x' \in \mathcal{X}$ and their images $z, z' \in \mathcal{Z}$ is at least $1-\mu$.

Proof.

$$\begin{aligned} & k_{zz'} - k_{xx'} \\ &= \sigma_y^2 \exp\left(-0.5\|z-z'\|^2/l^2\right) - \sigma_y^2 \exp\left(-0.5\|x-x'\|^2/l^2\right) \\ &\leq \sigma_y^2 \exp\left(-0.5(1-\nu)\|x-x'\|^2/l^2\right) - \sigma_y^2 \exp\left(-0.5\|x-x'\|^2/l^2\right) \\ &= k_{xx'}\left(\exp\left(0.5\nu\|x-x'\|^2/l^2\right) - 1\right) \\ &\leq k_{xx'}\left(2 \cdot \left(0.5\nu\|x-x'\|^2/l^2\right)\right) \\ &\leq k_{xx'} \cdot \nu\mathrm{d}^2 \end{aligned}$$

where the first inequality follows from Theorem 3.7 (since the condition $(1-\nu)\|x-x'\|^2 \leq \|z-z'\|^2$ holds in both cases $\sigma_{min}(\mathcal{X}) \geq \omega$ and otherwise), and the second inequality follows from the identity $\exp c \leq 1 + 2c$ for $c \in (0,1)$ by setting $c = 0.5\nu\|x-x'\|^2/l^2$ since $\nu \leq 2/\mathrm{d}^2$ and

$$\begin{aligned} & 0.5\nu\|x-x'\|^2/l^2 \\ &\leq 0.5\nu \ (\mathrm{diam}(\mathcal{X}))^2/l^2 \\ &\leq 0.5 \cdot 2/\mathrm{d}^2 \cdot (\mathrm{diam}(\mathcal{X}))^2/l^2 \\ &= 1. \end{aligned} \tag{A.10}$$

If $\sigma_{min}(\mathcal{X}) \geq \omega$,

$$\begin{aligned} & k_{xx'} - k_{zz'} \\ &= \sigma_y^2 \exp\left(-0.5\|x-x'\|^2/l^2\right) - \sigma_y^2 \exp\left(-0.5\|z-z'\|^2/l^2\right) \\ &\leq \sigma_y^2 \exp\left(-0.5\|x-x'\|^2/l^2\right) - \sigma_y^2 \exp\left(-0.5(1+\nu)\|x-x'\|^2/l^2\right) \\ &= k_{xx'}\left(1 - \exp\left(-0.5\nu\|x-x'\|^2/l^2\right)\right) \\ &= k_{xx'}\left(\exp\left(0.5\nu\|x-x'\|^2/l^2\right) - 1\right)\exp\left(-0.5\nu\|x-x'\|^2/l^2\right) \\ &\leq k_{xx'}\left(\exp\left(0.5\nu\|x-x'\|^2/l^2\right) - 1\right) \\ &\leq k_{xx'}\left(2 \cdot \left(0.5\nu\|x-x'\|^2/l^2\right)\right) \\ &\leq k_{xx'} \cdot \nu\mathrm{d}^2 \end{aligned}$$

where the first inequality follows from Theorem 3.7, since if $\sigma_{min}(\mathcal{X}) \geq \omega$, $C' = 1$ in the statement of Theorem 3.7, the second inequality follows from $0.5\nu\|x-x'\|^2/l^2 \geq 0$

and the third inequality follows from the identity $\exp c \leq 1+2c$ for $c \in (0,1)$ by setting $c = 0.5\nu\|x-x'\|^2/l^2$ and (A.10).

Similarly, if $\sigma_{min}(\mathcal{X}) < \omega$,

$$\begin{aligned}
&k_{xx'} - k_{zz'} \\
&= \sigma_y^2 \exp\left(-0.5\|x-x'\|^2/l^2\right) - \sigma_y^2 \exp\left(-0.5\|z-z'\|^2/l^2\right) \\
&\leq \sigma_y^2 \exp\left(-0.5\|x-x'\|^2/l^2\right) - \sigma_y^2 \exp\left(-0.5(1+\nu)(1+\omega^2/\sigma_{min}^2(\mathcal{X}))\|x-x'\|^2/l^2\right) \\
&= k_{xx'}\left(1 - \exp\left(-0.5(\nu + \nu\omega^2/\sigma_{min}^2(\mathcal{X}) + \omega^2/\sigma_{min}^2(\mathcal{X}))\|x-x'\|^2/l^2\right)\right) \\
&\leq k_{xx'}\left(1 - \exp\left(-0.5(\nu + \nu\omega^2/\sigma_{min}^2(\mathcal{X}) + \omega^2/\sigma_{min}^2(\mathcal{X}))\mathrm{d}^2\right)\right)
\end{aligned}$$

where the first inequality follows from Theorem 3.7, since if $\sigma_{min}(\mathcal{X}) < \omega$, $C' = 1 + \omega^2/\sigma_{min}^2(\mathcal{X})$ in the statement of Theorem 3.7.

□

A.5 Proof of Theorem 3.8

First we recall and introduce a few notations which we will use throughout this section. Let $\mathcal{X} \subset \mathbb{R}^d$ be a dataset and its image under Algorithm 2 be a dataset $\mathcal{Z} \subset \mathbb{R}^r$, $\mathbf{z}_{1:t-1} \triangleq \{z_1, \ldots, z_{t-1}\}$ be a set of transformed inputs selected by Algorithm 3 run on transformed dataset $\mathcal{Z}$ after $t-1$ iterations and the preimage of $\mathbf{z}_{1:t-1}$ under Algorithm 2 be a set $\mathbf{x}_{1:t-1} \triangleq \{x_1, \ldots, x_{t-1}\}$. Let $z \in \mathcal{Z}$ be an (unobserved) transformed input and $x \in \mathcal{X}$ be its preimage under Algorithm 2. Let f be a latent function sampled from a GP. Define

$$\begin{aligned}
&\tilde{f}(z) \triangleq f(x) \\
&\alpha_t(x, \mathbf{x}_{1:t-1}) \triangleq \mu_t(x) + \beta_t^{1/2}\sigma_t(x) \\
&\alpha_t(z, \mathbf{z}_{1:t-1}) \triangleq \tilde{\mu}_t(z) + \beta_t^{1/2}\tilde{\sigma}_t(z) \\
&z_t \triangleq \operatorname*{argmax}_{z \in \mathcal{Z}} \alpha_t(z, \mathbf{z}_{1:t-1}).
\end{aligned} \tag{A.11}$$

That is, $\tilde{f}$ is the latent function f defined over the transformed dataset $\mathcal{Z}$, $\alpha_t(z, \mathbf{z}_{1:t-1})$ is the function maximized by Algorithm 3 at iteration t, $\alpha_t(x, \mathbf{x}_{1:t-1})$ is the function maximized by GP-UCB algorithm run on the original dataset, z_t is the transformed input selected by Algorithm 3 at iteration t and x_t is the preimage of z_t under Algorithm 2.

Lemma A.2. *Let $\delta' \in (0,1)$ be given and $\beta_t \triangleq 2\log(nt^2\pi^2/6\delta')$. Then*

$$|f(x) - \mu_t(x)| \leq \beta_t^{1/2}\sigma_t(x) \quad \forall x \in \mathcal{X} \quad \forall t \in \mathbb{N}$$

holds with probability at least $1-\delta'$.

Proof. Lemma A.2 above corresponds to Lemma 5.1 in Srinivas *et al.* [2010]; see its proof therein. □

Lemma A.3. *Let $\delta' \in (0,1)$ be given and $\beta_t \triangleq 2\log(nt^2\pi^2/6\delta')$. Then the probability of*

$$\tilde{f}(z^*) - \tilde{f}(z_t) \leq 2\max_{x,z} |\alpha_t(z, \mathbf{z}_{1:t-1}) - \alpha_t(x, \mathbf{x}_{1:t-1})| + 2\beta_t^{1/2}\sigma_t(x_t)$$

for all $t \in \mathbb{N}$ is at least $1-\delta'$ where z^* is the maximizer of $\tilde{f}$ and $x \in \mathcal{X}$ is the preimage of $z \in \mathcal{Z}$ under Algorithm 2.

Proof.

$$\begin{aligned}
&\tilde{f}(z^*) - \tilde{f}(z_t) \\
&= f(x^*) - f(x_t) \\
&\leq \alpha_t(x^*, \mathbf{x}_{1:t-1}) - f(x_t) \\
&= \alpha_t(x^*, \mathbf{x}_{1:t-1}) - \alpha_t(z^*, \mathbf{z}_{1:t-1}) + \alpha_t(z^*, \mathbf{z}_{1:t-1}) - f(x_t) \\
&\leq \alpha_t(x^*, \mathbf{x}_{1:t-1}) - \alpha_t(z^*, \mathbf{z}_{1:t-1}) + \alpha_t(z_t, \mathbf{z}_{1:t-1}) - f(x_t) \\
&= \alpha_t(x^*, \mathbf{x}_{1:t-1}) - \alpha_t(z^*, \mathbf{z}_{1:t-1}) + \alpha_t(z_t, \mathbf{z}_{1:t-1}) - \alpha_t(x_t, \mathbf{x}_{1:t-1}) + \alpha_t(x_t, \mathbf{x}_{1:t-1}) - f(x_t) \\
&\leq 2\max_{x,z} |\alpha_t(z, \mathbf{z}_{1:t-1}) - \alpha_t(x, \mathbf{x}_{1:t-1})| + \alpha_t(x_t, \mathbf{x}_{1:t-1}) - f(x_t) \\
&\leq 2\max_{x,z} |\alpha_t(z, \mathbf{z}_{1:t-1}) - \alpha_t(x, \mathbf{x}_{1:t-1})| + 2\beta_t^{1/2}\sigma_t(x_t)
\end{aligned}$$

where the first equality is due to (A.11) and x^* is the maximizer of f, the first and the last inequalities are due to Lemma A.2 and the second inequality is due to the choice of z_t in (A.11). □

Lemma A.3 resembles Lemma 5.2 of Srinivas *et al.* [2010] with an added term $2\max_{x,z} |\alpha_t(z, \mathbf{z}_{1:t-1}) - \alpha_t(x, \mathbf{x}_{1:t-1})|$. It suggests that in order to bound regret $\tilde{f}(z^*) - \tilde{f}(z_t)$ incurred by Algorithm 3 at iteration t, we need to bound $|\alpha_t(z, \mathbf{z}_{1:t-1}) - \alpha_t(x, \mathbf{x}_{1:t-1})|$. Using the diagonal dominance assumption (Definition 3.9), we do it in the following two lemmas:

Lemma A.4. *Let $C > 0$ be given. If for all $x, x' \in \mathcal{X}$ and their images under Algorithm 2 $z, z' \in \mathcal{Z}$ holds $|k_{zz'} - k_{xx'}| \leq C \cdot k_{xx'}$, for all $t = 1, \ldots, T$ matrix $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$ is diagonally dominant, then for every unobserved transformed input $z \in \mathcal{Z}$ and its preimage under Algorithm 2 $x \in \mathcal{X}$*

$$|\tilde{\sigma}_t^2(z) - \sigma_t^2(x)| \leq C_1 / \sqrt{|\mathbf{x}_{1:t-1}|}$$

where

$$C_1 \triangleq C\sigma_y \sqrt{2\sigma_y^2 + \sigma_n^2} \Big(\sqrt{2}(1+C)^2 \sigma_y^2 / \sigma_n^2 + (2+C)C \Big).$$

Proof.

$$
\begin{aligned}
&|\tilde{\sigma}_t^2(z) - \sigma_t^2(x)| \\
&= |\big(k_{zz} - K_{z\mathbf{z}_{1:t-1}}(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{z}_{1:t-1}z}\big) \\
&\quad - \big(k_{xx} - K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{x}_{1:t-1}x}\big)| \\
&= |K_{z\mathbf{z}_{1:t-1}}(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{z}_{1:t-1}z} - K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{x}_{1:t-1}x}| \\
&\leq |K_{z\mathbf{z}_{1:t-1}}(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{z}_{1:t-1}z} - K_{z\mathbf{z}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{z}_{1:t-1}z}| \\
&+ |K_{z\mathbf{z}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{z}_{1:t-1}z} - K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{z}_{1:t-1}z}| \\
&+ |K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{z}_{1:t-1}z} - K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{x}_{1:t-1}x}| \\
&\leq (1+C)^2 \|K_{x\mathbf{x}_{1:t-1}}\| \cdot \sigma_y^2/\sigma_n^2 \cdot \sqrt{2}C/\sqrt{|\mathbf{x}_{1:t-1}|} + (2+C)C \cdot \|K_{x\mathbf{x}_{1:t-1}}\|/\sqrt{|\mathbf{x}_{1:t-1}|} \\
&= C\|K_{x\mathbf{x}_{1:t-1}}\|/\sqrt{|\mathbf{x}_{1:t-1}|}\Big(\sqrt{2}(1+C)^2\sigma_y^2/\sigma_n^2 + (2+C)C\Big) \\
&\leq C\sigma_y\sqrt{2\sigma_y^2 + \sigma_n^2}/\sqrt{|\mathbf{x}_{1:t-1}|}\Big(\sqrt{2}(1+C)^2\sigma_y^2/\sigma_n^2 + (2+C)C\Big)
\end{aligned}
\tag{A.12}
$$

where the first equality is due to (3.2), the second equality is due to $k_{xx} = k_{zz} = \sigma_y^2$ for every x and z, the first inequality is due to triangle inequality, the second inequality is due to

$$
\begin{aligned}
&|K_{z\mathbf{z}_{1:t-1}}(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{z}_{1:t-1}z} - K_{z\mathbf{z}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{z}_{1:t-1}z}| \\
&= |K_{z\mathbf{z}_{1:t-1}}\big((K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1} - (K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\big) K_{\mathbf{z}_{1:t-1}z}| \\
&\leq \|K_{z\mathbf{z}_{1:t-1}}\|^2 \cdot \|(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1} - (K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \\
&\leq (1+C)^2 \|K_{x\mathbf{x}_{1:t-1}}\|^2 \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} - (K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \\
&\leq (1+C)^2 \|K_{x\mathbf{x}_{1:t-1}}\|^2 \cdot \|(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1}(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}})\|_2 \\
&\quad \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \\
&\leq (1+C)^2 \|K_{x\mathbf{x}_{1:t-1}}\|^2 \cdot \|(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \cdot \|K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}\|_2 \\
&\quad \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \\
&\leq (1+C)^2 \|K_{x\mathbf{x}_{1:t-1}}\|^2 \cdot 1/\sigma_n^2 \cdot \|K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}\|_2 \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \\
&\leq (1+C)^2 \|K_{x\mathbf{x}_{1:t-1}}\|^2 \cdot 1/\sigma_n^2 \cdot \sqrt{2}C\sigma_y^2/\sqrt{|\mathbf{x}_{1:t-1}|} \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \\
&\leq (1+C)^2 \|K_{x\mathbf{x}_{1:t-1}}\|^2 \cdot 1/\sigma_n^2 \cdot \sqrt{2}C\sigma_y^2/\sqrt{|\mathbf{x}_{1:t-1}|} \cdot 1/(\sqrt{|\mathbf{x}_{1:t-1}|}\|K_{x\mathbf{x}_{1:t-1}}\|) \\
&= (1+C)^2 \|K_{x\mathbf{x}_{1:t-1}}\| \cdot \sigma_y^2/\sigma_n^2 \cdot \sqrt{2}C/|\mathbf{x}_{1:t-1}| \\
&\leq (1+C)^2 \|K_{x\mathbf{x}_{1:t-1}}\| \cdot \sigma_y^2/\sigma_n^2 \cdot \sqrt{2}C/\sqrt{|\mathbf{x}_{1:t-1}|}
\end{aligned}
$$

where the first inequality is due to property of quadratic forms $|v^\top A v| \leq \|v\|^2 \cdot \|A\|_2$ for any vector v (see Theorem 2.11, Section II.2.2 in Stewart and Sun [1990]), the second inequality follows from the statement of the lemma and Remark A.1 to Theorem A.1, the third inequality follows from Theorem 2.5 (see Section III.2.2 in Stewart and Sun [1990]), the fourth inequality is due to the submultiplicativity of the spectral norm (see Section II.2.2, p. 69 in Stewart and Sun [1990]), the fifth inequality follows from Lemma A.7, the sixth inequality follows from Lemma A.8, the second last inequality follows from Lemma A.9 and the last inequality follows from $|\mathbf{x}_{1:t-1}| \geq 1$;

and

$$
\begin{aligned}
&|K_{z\mathbf{z}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1}K_{\mathbf{z}_{1:t-1}z} - K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1}K_{\mathbf{z}_{1:t-1}z}| \\
&+ |K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1}K_{\mathbf{z}_{1:t-1}z} - K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1}K_{\mathbf{x}_{1:t-1}x}| \\
&= |(K_{z\mathbf{z}_{1:t-1}} - K_{x\mathbf{x}_{1:t-1}})(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1}K_{\mathbf{z}_{1:t-1}z}| \\
&+ |K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1}(K_{\mathbf{z}_{1:t-1}z} - K_{\mathbf{x}_{1:t-1}x})| \\
&\leq \|K_{z\mathbf{z}_{1:t-1}} - K_{x\mathbf{x}_{1:t-1}}\| \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1}\|_2 \cdot \|K_{\mathbf{z}_{1:t-1}z}\| \\
&+ \|K_{x\mathbf{x}_{1:t-1}}\| \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1}\|_2 \cdot \|K_{\mathbf{z}_{1:t-1}z} - K_{\mathbf{x}_{1:t-1}x}\| \\
&\leq (1+1+C)\cdot \|K_{z\mathbf{z}_{1:t-1}} - K_{x\mathbf{x}_{1:t-1}}\| \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1}\|_2 \cdot \|K_{\mathbf{x}_{1:t-1}x}\| \\
&\leq (2+C)\cdot C\|K_{x\mathbf{x}_{1:t-1}}\| \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1}\|_2 \cdot \|K_{x\mathbf{x}_{1:t-1}}\| \\
&\leq (2+C)\cdot C\|K_{x\mathbf{x}_{1:t-1}}\| \cdot 1/(\sqrt{|\mathbf{x}_{1:t-1}|}\|K_{x\mathbf{x}_{1:t-1}}\|) \cdot \|K_{x\mathbf{x}_{1:t-1}}\| \\
&= (2+C)C \cdot \|K_{x\mathbf{x}_{1:t-1}}\|/\sqrt{|\mathbf{x}_{1:t-1}|}
\end{aligned}
$$

where the first inequality is due to property of bilinear forms $|u^\top A v| \leq \|u\| \cdot \|A\|_2 \cdot \|v\|$ for any vectors u, v (see Theorem 2.11, Section II.2.2 in Stewart and Sun [1990]), the second and the third inequalities follow from the statement of the lemma and Remark A.1 to Theorem A.1 and the last inequality follows from Lemma A.9.

The last inequality in (A.12) follows from

$$
\begin{aligned}
&\|K_{x\mathbf{x}_{1:t-1}}\|^2 \\
&= \|K_{x\mathbf{x}_{1:t-1}}\|^2 \cdot \psi_{max}^{-1}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)\cdot \psi_{max}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I) \\
&= \|K_{x\mathbf{x}_{1:t-1}}\|^2 \cdot \psi_{min}((K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1})\cdot \psi_{max}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I) \\
&= \|K_{x\mathbf{x}_{1:t-1}}\|^2 \cdot \psi_{min}((K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1})\cdot \|K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I\|_2 \\
&= \|K_{x\mathbf{x}_{1:t-1}}\|^2 \cdot \psi_{min}((K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1})\cdot (\|K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}\|_2+\sigma_n^2) \\
&\leq \|K_{x\mathbf{x}_{1:t-1}}\|^2 \cdot \psi_{min}((K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1})\cdot (2\sigma_y^2+\sigma_n^2) \\
&\leq K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}+\sigma_n^2 I)^{-1}K_{\mathbf{x}_{1:t-1}x}\cdot (2\sigma_y^2+\sigma_n^2) \\
&\leq k_{xx}\cdot(2\sigma_y^2+\sigma_n^2) \\
&= \sigma_y^2(2\sigma_y^2+\sigma_n^2)
\end{aligned}
$$

where $\psi_{max}(\cdot)$ and $\psi_{min}(\cdot)$ denote the largest and the smallest eigenvalues of a matrix, respectively, the first fourth equalities are properties of eigenvalues, the first inequality is due to Lemma A.10, the second inequality follows from Lemma A.11, the third inequality follows from the fact that conditioning does not increase variance and the last equality is due to $k_{xx} = \sigma_y^2$. □

Lemma A.5. *Let $C > 0$ be given. If for all $x, x' \in \mathcal{X}$ and their images under Algorithm 2 $z, z' \in \mathcal{Z}$ holds $|k_{zz'} - k_{xx'}| \leq C \cdot k_{xx'}$, for all $t = 1, \ldots, T$ matrix $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$ is diagonally dominant and $|y_t| \leq L$, then for every unobserved transformed input $z \in \mathcal{Z}$ and its preimage under Algorithm 2 $x \in \mathcal{X}$*

$$|\tilde{\mu}_t(z) - \mu_t(x)| \leq CL + C_2/\sqrt{|\mathbf{x}_{1:t-1}|}$$

where

$$C_2 = \sqrt{2}(1+C)\cdot C\sigma_y^2/\sigma_n^2 \cdot L.$$

Proof.

$$\begin{aligned}
&|\tilde{\mu}_t(z) - \mu_t(x)| \\
&= |K_{z\mathbf{z}_{1:t-1}}(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1}\mathbf{y}_{t-1} - K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\mathbf{y}_{t-1}| \\
&\leq |K_{z\mathbf{z}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\mathbf{y}_{t-1} - K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\mathbf{y}_{t-1}| \\
&+ |K_{z\mathbf{z}_{1:t-1}}(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1}\mathbf{y}_{t-1} - K_{z\mathbf{z}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\mathbf{y}_{t-1}| \\
&= |(K_{z\mathbf{z}_{1:t-1}} - K_{x\mathbf{x}_{1:t-1}})(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\mathbf{y}_{t-1}| \\
&+ |K_{z\mathbf{z}_{1:t-1}}\big((K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1} - (K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\big)\mathbf{y}_{t-1}| \\
&\leq C\cdot L + C_2/\sqrt{|\mathbf{x}_{1:t-1}|}
\end{aligned}$$

where the first equality is due to (3.2), the first inequality is due to triangle inequality and the second inequality follows from

$$\begin{aligned}
&|(K_{z\mathbf{z}_{1:t-1}} - K_{x\mathbf{x}_{1:t-1}})(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\mathbf{y}_{t-1}| \\
&\leq \|K_{z\mathbf{z}_{1:t-1}} - K_{x\mathbf{x}_{1:t-1}}\| \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \cdot \|\mathbf{y}_{t-1}\| \\
&\leq C\|K_{x\mathbf{x}_{1:t-1}}\| \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \cdot \|\mathbf{y}_{t-1}\| \\
&\leq C\|K_{x\mathbf{x}_{1:t-1}}\| \cdot 1/(\sqrt{|\mathbf{x}_{1:t-1}|}\|K_{x\mathbf{x}_{1:t-1}}\|) \cdot \|\mathbf{y}_{t-1}\| \\
&\leq C\cdot L
\end{aligned}$$

where the first inequality is due to property of bilinear forms $|u^\top A v| \leq \|u\|\cdot\|A\|_2\cdot\|v\|$ for any vectors u, v (see Theorem 2.11, Section II.2.2 in Stewart and Sun [1990]), the second inequality follows from the statement of the lemma, the third inequality follows from Lemma A.9 and the last inequality follows from the condition $|y_t| \leq L$ for all $t = 1, \ldots, T$;

and

$$\begin{aligned}
&|K_{z\mathbf{z}_{1:t-1}}\big((K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1} - (K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\big)\mathbf{y}_{t-1}| \\
&\leq \|K_{z\mathbf{z}_{1:t-1}}\| \cdot \|(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1} - (K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \cdot \|\mathbf{y}_{t-1}\| \\
&\leq \|K_{z\mathbf{z}_{1:t-1}}\| \cdot \|(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \\
&\quad \cdot \|(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}})(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \cdot \|\mathbf{y}_{t-1}\| \\
&\leq \|K_{z\mathbf{z}_{1:t-1}}\| \cdot \|(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \\
&\quad \cdot \|K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}\|_2 \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \cdot \|\mathbf{y}_{t-1}\| \\
&\leq \|K_{z\mathbf{z}_{1:t-1}}\| \cdot 1/\sigma_n^2 \cdot \|K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}\|_2 \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \cdot \|\mathbf{y}_{t-1}\| \\
&\leq \|K_{z\mathbf{z}_{1:t-1}}\| \cdot 1/\sigma_n^2 \cdot \sqrt{2}C\sigma_y^2/\sqrt{|\mathbf{x}_{1:t-1}|} \cdot \|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \cdot \|\mathbf{y}_{t-1}\| \\
&\leq \|K_{z\mathbf{z}_{1:t-1}}\| \cdot 1/\sigma_n^2 \cdot \sqrt{2}C\sigma_y^2/\sqrt{|\mathbf{x}_{1:t-1}|} \cdot 1/(\sqrt{|\mathbf{x}_{1:t-1}|}\|K_{x\mathbf{x}_{1:t-1}}\|) \cdot \|\mathbf{y}_{t-1}\| \\
&\leq (1+C)\|K_{x\mathbf{x}_{1:t-1}}\| \cdot 1/\sigma_n^2 \cdot \sqrt{2}C\sigma_y^2/\sqrt{|\mathbf{x}_{1:t-1}|} \cdot 1/(\sqrt{|\mathbf{x}_{1:t-1}|}\|K_{x\mathbf{x}_{1:t-1}}\|) \cdot \|\mathbf{y}_{t-1}\| \\
&\leq \sqrt{2}(1+C)\cdot C\sigma_y^2/\sigma_n^2 \cdot L/\sqrt{|\mathbf{x}_{1:t-1}|} \\
&= C_2/\sqrt{|\mathbf{x}_{1:t-1}|}
\end{aligned}$$

where the first inequality is due to property of bilinear forms $|u^\top Av| \leq \|u\| \cdot \|A\|_2 \cdot \|v\|$ for any vectors u, v (see Theorem 2.11, Section II.2.2 in Stewart and Sun [1990]), the second inequality follows from Theorem 2.5 (see Section III.2.2 in Stewart and Sun [1990]), the third inequality is due to the submultiplicativity of the spectral norm (see Section II.2.2, p. 69 in Stewart and Sun [1990]) the fourth inequality follows from Lemma A.7, the fifth inequality follows from Lemma A.8, the third last inequality follows from Lemma A.9, the second last inequality follows from the statement of the lemma and Remark A.1 to Theorem A.1 and the last inequality follows from the condition $|y_t| \leq L$ for all $t = 1, \ldots, T$. □

Proof of the theorem. By Lemma A.3 for $\delta' = \delta_{ucb}/2$ and $\beta_t = 2\log(nt^2\pi^2/3\delta_{ucb})$ for all $t \in \mathbb{N}$:

$$\begin{aligned} & r_t \\ & = f(x^*) - f(x_t) \\ & = \tilde{f}(z^*) - \tilde{f}(z_t) \\ & \leq 2 \max_{x,z} |\alpha_t(z, \mathbf{z}_{1:t-1}) - \alpha_t(x, \mathbf{x}_{1:t-1})| + 2\beta_t^{1/2}\sigma_t(x_t) \\ & \leq 2 \max_{x,z} |\tilde{\mu}_t(z) - \mu_t(x)| + 2\beta_t^{1/2} \max_{x,z} |\tilde{\sigma}_t^2(z) - \sigma_t^2(x)| + 2\beta_t^{1/2}\sigma_t(x_t) \end{aligned} \tag{A.13}$$

with probability at least $1 - \delta_{ucb}/2$ where the second equality follows from (A.11), the first inequality follows from Lemma A.3 and the second inequality follows from triangle inequality. Suppose $\nu \in (0, \min(1/2, 2/\mathrm{d}^2))$, $\mu \in (0, 1)$ are given (we will set the exact values of μ, ν later) and the input parameter of Algorithm 2 $r \geq 8\log(n^2/\mu)/\nu^2$. By Theorem A.1 for all $x, x' \in \mathcal{X}$ and their images under Algorithm 2 $z, z' \in \mathcal{Z}$ holds $|k_{zz'} - k_{xx'}| \leq C \cdot k_{xx'}$ with probability at least $1 - \mu$. Let $\mu = \delta_{ucb}/2$. Then we can apply Lemma A.4 and Lemma A.5 to (A.13). Using the union bound we obtain that for all $t = 1, \ldots, T$

$$\begin{aligned} & r_t \\ & \leq 2 \max_{x,z} |\tilde{\mu}_t(z) - \mu_t(x)| + 2\beta_t^{1/2} \max_{x,z} |\tilde{\sigma}_t^2(z) - \sigma_t^2(x)| + 2\beta_t^{1/2}\sigma_t(x_t) \\ & \leq 2(CL + C_2/\sqrt{|\mathbf{x}_{1:t-1}|}) + 2C_1\beta_t^{1/2}/\sqrt{|\mathbf{x}_{1:t-1}|} + 2\beta_t^{1/2}\sigma_t(x_t) \end{aligned} \tag{A.14}$$

with probability at least $1 - \delta_{ucb}$ where C_1 and C_2 are defined in Lemma A.4 and

Lemma A.5, respectively. Summing over $t = 1, \ldots, T$:

$$\begin{aligned}
&\sum_{t=1}^{T} r_t^2 \\
&\leq 4 \sum_{t=1}^{T} \left(CL + C_2 / \sqrt{|\mathbf{x}_{1:t-1}|} + C_1 \beta_t^{1/2} / \sqrt{|\mathbf{x}_{1:t-1}|} + \beta_t^{1/2} \sigma_t(x_t) \right)^2 \\
&\leq 12 \sum_{t=1}^{T} \left(C^2 L^2 + (C_2 + C_1 \beta_t^{1/2})^2 / |\mathbf{x}_{1:t-1}| + \beta_t \sigma_t^2(x_t) \right) \\
&= 12 C^2 L^2 T + 12 \sum_{t=1}^{T} (C_2 + C_1 \beta_t^{1/2})^2 |\mathbf{x}_{1:t-1}| + 12 \sum_{t=1}^{T} \beta_t \sigma_t^2(x_t) \\
&\leq 12 C^2 L^2 T + 24 (C_2 + C_1 \beta_T^{1/2})^2 \log T + 12 \beta_T \sum_{t=1}^{T} \sigma_t^2(x_t) \\
&\leq 12 C^2 L^2 T + 24 (C_2 + C_1 \beta_T^{1/2})^2 \log T + 12 \beta_T / \log(1 + \sigma_n^{-2}) \sum_{t=1}^{T} \log(1 + \sigma_n^{-2} \sigma_t^2(x_t)) \\
&\leq 12 C^2 L^2 T + 24 (C_2 + C_1 \beta_T^{1/2})^2 \log T + 24 \beta_T / \log(1 + \sigma_n^{-2}) \cdot \gamma_T
\end{aligned} \tag{A.15}$$

where the first inequality follows from (A.14), the second inequality follows from identity $(a+b+c)^2 \leq 3(a^2+b^2+c^2)$, the third inequality follows from $\sum_{t=1}^{T} 1/|\mathbf{x}_{1:t-1}| \leq \sum_{t=1}^{T} 1/t \leq 2 \log T$ and the fact that β_t is nondecreasing, the fourth inequality corresponds to an intermediate step of Lemma 5.4 in Srinivas *et al.* [2010] and the last step follows from Lemma 5.3 and Lemma 5.4 in Srinivas *et al.* [2010] where $\gamma_T \triangleq \max_{\mathbf{x}_{1:T} \subset \mathcal{X}} \mathbb{I}[f_{\mathcal{X}}; \mathbf{y}_{1:T}] = \mathcal{O}\big((\log T)^{d+1}\big)$ and $f_{\mathcal{X}} \triangleq \{f(x)\}_{x \in \mathcal{X}}$ (see Theorem 5 in Srinivas *et al.* [2010]). Therefore,

$$\begin{aligned}
&S_T^2 \\
&\leq R_T^2 / T^2 \\
&\leq \sum_{t=1}^{T} r_t^2 / T \\
&\leq 12 C^2 L^2 + 24 (C_2 + C_1 \beta_T^{1/2})^2 \log T / T + 24 \beta_T / \log(1 + \sigma_n^{-2}) \gamma_T / T
\end{aligned} \tag{A.16}$$

where the second inequality follows from Cauchy-Schwarz inequality and the last inequality follows from (A.15). If $\sigma_{min}(\mathcal{X}) \geq \omega$ then, according to Theorem A.1, $C = \nu \mathrm{d}^2$. To guarantee that $12 C^2 L^2 \leq \epsilon_{ucb}^2$ and to satisfy the premise of Lemma 3.1 (i.e. $\nu \leq 1/2$) and Theorem A.1 (i.e. $\nu \leq 2/\mathrm{d}^2$), we need to set the value of $\nu = \min(\varepsilon_{ucb} / (2\sqrt{3} \mathrm{d}^2 L), 2/\mathrm{d}^2, 1/2)$.

Since $\nu \leq 2/\mathrm{d}^2$ and hence $C = \nu \mathrm{d}^2 \leq 2$

$$\begin{aligned}
&C_1 \\
&= C\sigma_y \sqrt{2\sigma_y^2 + \sigma_n^2} \Big(\sqrt{2}(1+C)^2 \sigma_y^2/\sigma_n^2 + (2+C)C \Big) \\
&\leq 2\sigma_y \sqrt{2\sigma_y^2 + \sigma_n^2} \Big(\sqrt{2}(1+2)^2 \sigma_y^2/\sigma_n^2 + (2+2)\cdot 2 \Big) \\
&= \mathcal{O}\Big(\sigma_y \sqrt{\sigma_y^2 + \sigma_n^2} (\sigma_y^2/\sigma_n^2 + 1) \Big)
\end{aligned}$$

and

$$\begin{aligned}
&C_2 \\
&= \sqrt{2}(1+C) \cdot C\sigma_y^2/\sigma_n^2 \cdot L \\
&\leq \sqrt{2}(1+2) \cdot 2\sigma_y^2/\sigma_n^2 \cdot L \\
&= \mathcal{O}(\sigma_y^2/\sigma_n^2 \cdot L)
\end{aligned}$$

where C_1 and C_2 are defined in Lemma A.4 and Lemma A.5, respectively.

Remark A.2. If $\sigma_{min}(\mathcal{X}) < \omega$, a similar form of regret bound to that of (A.16) can be proven: According to Theorem A.1,

$$C = \max(\nu \mathrm{d}^2, 1 - \exp\left(-0.5(\nu + \nu\omega^2/\sigma_{min}^2(\mathcal{X}) + \omega^2/\sigma_{min}^2(\mathcal{X}))\mathrm{d}^2\right))$$

instead of $C = \nu \mathrm{d}^2$ and the entire proof of Theorem 3.8 can be directly copied to reach (A.16). In this case, however, the term $12C^2L^2$ in (A.16) cannot be set arbitrarily small. That is explained by the fact that when $\sigma_{min}(\mathcal{X}) < \omega$, Algorithm 2 increases the singular values of dataset $\mathcal{X}$ (see line 9) and the pairwise distances between the original inputs from $\mathcal{X}$ are no longer approximately the same as the distances between their respective transformed images (see Theorem 3.7) resulting in a looser regret bound.

A.6 Auxiliary results

Lemma A.6. *Let a dataset $\mathcal{X} \subset \mathbb{R}^d$ be given. Let a dataset $\tilde{\mathcal{X}} \subset \mathbb{R}^d$ be defined in line 9 of Algorithm 2 (i.e., $\tilde{\mathcal{X}} = U\sqrt{\Sigma^2 + \omega^2 I_{n\times d}}V^\top$ where $\mathcal{X} = U\Sigma V^\top$ is the singular value decomposition of $\mathcal{X}$). Let $\sigma_{min}(\mathcal{X}) > 0$ be the smallest singular value of $\mathcal{X}$. Then for all $x, x' \in \mathcal{X}$ and their corresponding $\tilde{x}, \tilde{x}' \in \tilde{\mathcal{X}}$ (when viewing datasets $\mathcal{X}$ and $\tilde{\mathcal{X}}$ as matrices)*

$$\|x - x'\| \leq \|\tilde{x} - \tilde{x}'\| \leq \sqrt{1 + \omega^2/\sigma_{min}^2(\mathcal{X})}\|x - x'\|.$$

Proof. Denote the rows of U as $u_{(i)}$ so that

$$U = \begin{bmatrix} u_{(1)} \\ \vdots \\ u_{(n)} \end{bmatrix}.$$

For $i = 1, \ldots, n$ denote the input in the i-th row of the datset $\mathcal{X}$ ($\tilde{\mathcal{X}}$) viewed as matrix as $x_{(i)}$ ($\tilde{x}_{(i)}$). From the singular value decomposition, $x_{(i)} = u_{(i)}\Sigma V^\top$ and $\tilde{x}_{(i)} = u_{(i)}\sqrt{\Sigma^2 + I_{n\times d}\omega^2}V^\top$ Then for $i, j = 1, \ldots, n$

$$\begin{aligned}
&\|\tilde{x}_{(i)} - \tilde{x}_{(j)}\|^2 \\
&= \|(u_{(i)} - u_{(j)})\sqrt{\Sigma^2 + \omega^2 I_{n\times d}}V^\top\|^2 \\
&= (u_{(i)} - u_{(j)})\sqrt{\Sigma^2 + \omega^2 I_{n\times d}}V^\top V\sqrt{\Sigma^2 + \omega^2 I_{n\times d}}^\top (u_{(i)} - u_{(j)})^\top \\
&= (u_{(i)} - u_{(j)})\sqrt{\Sigma^2 + \omega^2 I_{n\times d}}\sqrt{\Sigma^2 + \omega^2 I_{n\times d}}^\top (u_{(i)} - u_{(j)})^\top \\
&= \sum_{k=1}^{\min(n,d)} (u_{(i)k} - u_{(j)k})^2(\sigma_k^2 + \omega^2) \\
&\leq \sum_{k=1}^{\min(n,d)} (u_{(i)k} - u_{(j)k})^2\sigma_k^2(1 + \omega^2/\sigma_{min}^2(\mathcal{X})) \\
&= (1 + \omega^2/\sigma_{min}^2(\mathcal{X}))(u_{(i)} - u_{(j)})\Sigma\Sigma^\top(u_{(i)} - u_{(j)})^\top \\
&= (1 + \omega^2/\sigma_{min}^2(\mathcal{X}))(u_{(i)} - u_{(j)})\Sigma V^\top V\Sigma^\top(u_{(i)} - u_{(j)})^\top \\
&= (1 + \omega^2/\sigma_{min}^2(\mathcal{X}))\|(u_{(i)} - u_{(j)})\Sigma V^\top\|^2 \\
&= (1 + \omega^2/\sigma_{min}^2(\mathcal{X}))\|x_{(i)} - x_{(j)}\|^2
\end{aligned} \tag{A.17}$$

where the second and the second last equalities follow from $\|v\|^2 = vv\top$ for any row vector v, the third and the third last equalities follow from orthonormality of matrix V, and the inequality follows from

$$\begin{aligned}
&\sigma_k^2 + \omega^2 \\
&= \sigma_k^2(1 + \omega^2/\sigma_k^2) \\
&\leq \sigma_k^2(1 + \omega^2/\sigma_{min}^2(\mathcal{X}))
\end{aligned}$$

where the inequality follows from $\sigma_k \geq \sigma_{min}(\mathcal{X})$ for every $k = 1, \ldots, \min(n, d)$.

Similarly,

$$\begin{aligned}
&\|\tilde{x}_{(i)} - \tilde{x}_{(j)}\|^2 \\
&= \sum_{k=1}^{\min(n,d)} (u_{(i)k} - u_{(j)k})^2(\sigma_k^2 + \omega^2) \\
&= \sum_{k=1}^{\min(n,d)} (u_{(i)k} - u_{(j)k})^2\sigma_k^2 + \omega^2 \sum_{k=1}^{\min(n,d)} (u_{(i)k} - u_{(j)k})^2 \qquad \text{(A.18)} \\
&\geq \sum_{k=1}^{\min(n,d)} (u_{(i)k} - u_{(j)k})^2\sigma_k^2 \\
&= \|x_{(i)} - x_{(j)}\|^2
\end{aligned}$$

where the first and the last equalities follow from the fourth and the fifth equalities of (A.17), respectively. Since (A.17) and (A.18) both hold for all $i, j = 1, \ldots, n$, the lemma follows. □

Lemma A.7. *In the notations of Section A.5, for all $t = 1, \ldots, T$ holds $\|(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \leq 1/\sigma_n^2$.*

Proof. Since $(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1}$ is positive definite, by definition of spectral norm for all $t = 1, \ldots, T$ and $\mathbf{z}_{1:t-1}$

$$\begin{aligned}
&\|(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \\
&= \psi_{max}((K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)^{-1}) \\
&= \frac{1}{\psi_{min}(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} + \sigma_n^2 I)} \\
&= \frac{1}{\psi_{min}(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}}) + \sigma_n^2} \\
&\leq 1/\sigma_n^2
\end{aligned}$$

where $\psi_{max}(\cdot)$ and $\psi_{min}(\cdot)$ denote the largest and the smallest eigenvalues of a matrix, respectively, the second and the third equalities are properties of eigenvalues and the inequality is due to the fact that matrix $K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}}$ is positive semidefinite. □

Lemma A.8. *In the notations of Section A.5, if for all $x, x' \in \mathcal{X}$ and their images under Algorithm 2 $z, z' \in \mathcal{Z}$ holds $|k_{zz'} - k_{xx'}| \leq C \cdot k_{xx'}$, and for all $t = 1, \ldots, T$ matrix $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$ is diagonally dominant (Definition 3.9), then*

$$\|K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}\|_2 \leq \sqrt{2}C\sigma_y^2/\sqrt{|\mathbf{x}_{1:t-1}|}.$$

Proof. Fix $t = 1, \ldots, T$. For some $i = 1, \ldots, t-1$:

$$
\begin{aligned}
&\|K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}\|_2^2 \\
&= \psi_{max}\big((K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}})^\top (K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}})\big) \\
&= \psi_{max}\big((K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}})^2\big) \\
&\leq \sum_{j,j\neq i} |[(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}})^2]_{ij}| + [(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}})^2]_{ii} \\
&\leq 2C^2\sigma_y^4/\big(\sqrt{|\mathbf{x}_{1:t-1}| - 1} + 1\big)^2 \\
&\leq 2C^2\sigma_y^4/|\mathbf{x}_{1:t-1}|
\end{aligned}
$$

where $\psi_{max}(\cdot)$ denotes the largest eigenvalue of a matrix, the first equality is the definition of spectral norm, the second equality follows from the fact that matrices $K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}}$ and $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$ are symmetric, the first inequality is due to Gershgorin circle theorem, the last inequality follows from $\sqrt{|\mathbf{x}_{1:t-1}| - 1} + 1 \geq \sqrt{|\mathbf{x}_{1:t-1}|}$ and the second last inequality follows from

$$
\begin{aligned}
&\sum_{j,j\neq i} |[(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}})^2]_{ij}| \\
&= \sum_{j,j\neq i} |\sum_{p} [K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}]_{ip} [K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}]_{pj}| \\
&= \sum_{j,j\neq i} |\sum_{p} (k_{z_i z_p} - k_{x_i x_p})(k_{z_p z_j} - k_{x_p x_j})| \\
&= \sum_{j,j\neq i} |\sum_{p,p\neq j,i} (k_{z_i z_p} - k_{x_i x_p})(k_{z_p z_j} - k_{x_p x_j})| \\
&\leq \sum_{j,j\neq i} \sum_{p,p\neq j,i} |k_{z_i z_p} - k_{x_i x_p}| \cdot |k_{z_p z_j} - k_{x_p x_j}| \\
&\leq C^2 \sum_{j,j\neq i} \sum_{p,p\neq j} k_{x_i x_p} \cdot k_{x_p x_j} \\
&= C^2 \sum_{p,p\neq j,i} k_{x_i x_p} \sum_{j,j\neq i,p} k_{x_p x_j} \\
&\leq C^2 \sum_{p,p\neq j,i} k_{x_i x_p} k_{x_p x_p} / \big(\sqrt{|\mathbf{x}_{1:t-1}| - 1} + 1\big) \\
&= C^2\sigma_y^2/\big(\sqrt{|\mathbf{x}_{1:t-1}| - 1} + 1\big) \sum_{p,p\neq j,i} k_{x_i x_p} \\
&\leq C^2\sigma_y^2/\big(\sqrt{|\mathbf{x}_{1:t-1}| - 1} + 1\big) k_{x_i x_i} / \big(\sqrt{|\mathbf{x}_{1:t-1}| - 1} + 1\big) \\
&= C^2\sigma_y^4/\big(\sqrt{|\mathbf{x}_{1:t-1}| - 1} + 1\big)^2
\end{aligned}
$$

where the third, the fifth and the last equalities follow from $k_{z_p z_p} = k_{x_p x_p} = \sigma_y^2$ for every p, the first inequality follows from triangle inequality, the second inequality follows from the statement of the lemma, the third and the last inequalities follow

from the diagonal dominance property of $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$ (Definition 3.9); and

$$
\begin{aligned}
&[(K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}})^2]_{ii} \\
&= \sum_p [K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}]_{ip} [K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}]_{pi} \\
&= \sum_p [K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}} - K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}]_{ip}^2 \\
&= \sum_p (k_{z_i z_p} - k_{x_i x_p})^2 \\
&= \sum_{p, p\neq i} (k_{z_i z_p} - k_{x_i x_p})^2 \\
&\leq C^2 \sum_{p, p\neq i} k_{x_i x_p}^2 \\
&\leq C^2 \big(\sum_{p, p\neq i} k_{x_i x_p} \big)^2 \\
&\leq C^2 k_{x_i x_i}^2 / \big(\sqrt{|\mathbf{x}_{1:t-1}| - 1} + 1 \big)^2 \\
&= C^2 \sigma_y^4 / \big(\sqrt{|\mathbf{x}_{1:t-1}| - 1} + 1 \big)^2
\end{aligned}
$$

where the second equality follows from the fact that $K_{\mathbf{z}_{1:t-1}\mathbf{z}_{1:t-1}}$ and $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$ are symmetric, the fourth and the last equalities follow from $k_{z_p z_p} = k_{x_p x_p} = \sigma_y^2$ for every p, the first inequality follows from the statement of the lemma and the last inequality follows from the diagonal dominance of $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$ (Definition 3.9). □

Lemma A.9. *In the notations of Section A.5, if for all $t = 1, \ldots, T$ matrix $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$ is diagonally dominant (Definition 3.9), then for any unobserved original input $x \in \mathcal{X}$ at iteration t*

$$
\|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \leq 1/(\sqrt{|\mathbf{x}_{1:t-1}|}\|K_{x\mathbf{x}_{1:t-1}}\|).
$$

.

Proof. By applying Gershgorin circle theorem for $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$:

$$
\begin{aligned}
&\psi_{min}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}) \\
&\geq \min_{x_i \in \mathbf{x}_{1:t-1}} \big(k_{x_i x_i} - R_{\mathbf{x}_{1:t-1}}(x_i) \big) \\
&= k_{xx} - \max_{x_i \in \mathbf{x}_{1:t-1}} R_{\mathbf{x}_{1:t-1}}(x_i) \\
&\geq (\sqrt{|\mathbf{x}_{1:t-1}|} + 1) \max_{x_i \in \mathbf{x}_{1:t-1} \cup \{x\}} R_{\mathbf{x}_{1:t-1}\cup\{x\}}(x_i) - \max_{x_i \in \mathbf{x}_{1:t-1}} R_{\mathbf{x}_{1:t-1}}(x_i)
\end{aligned}
$$

where $\psi_{min}(\cdot)$ denotes the smallest eigenvalue of a matrix, $R_{\mathbf{x}_{1:t-1}}(x_i) \triangleq \sum_{x_j \in \mathbf{x}_{1:t-1}\setminus\{x_i\}} k_{x_i x_j}$, the first equality follows from the fact that $k_{xx} = \sigma_y^2 = k_{x_i x_i}$ for all x_i and x, and

the second inequality holds because $K_{(\mathbf{x}_{1:t-1}\cup\{x\})(\mathbf{x}_{1:t-1}\cup\{x\})}$ is assumed to be diagonally dominant. On the other hand, since $x \notin \mathbf{x}_{1:t-1}$, $R_{\mathbf{x}_{1:t-1}\cup\{x\}}(x_i) = R_{\mathbf{x}_{1:t-1}}(x_i) + k_{x_i x}$ for all $x_i \in \mathbf{x}_{1:t-1}$, which immediately implies $\max_{x_i\in\mathbf{x}_{1:t-1}\cup\{x\}} R_{\mathbf{x}_{1:t-1}\cup\{x\}}(x_i) \geq \max_{x_i\in\mathbf{x}_{1:t-1}} R_{\mathbf{x}_{1:t-1}\cup\{x\}}(x_i) \geq \max_{x_i\in\mathbf{x}_{1:t-1}} R_{\mathbf{x}_{1:t-1}}(x_i)$. Plugging this into above inequality,

$$\begin{aligned}
&\psi_{min}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}) \\
&\geq (\sqrt{|\mathbf{x}_{1:t-1}|}+1) \max_{x_i\in\mathbf{x}_{1:t-1}\cup\{x\}} R_{\mathbf{x}_{1:t-1}\cup\{x\}}(x_i) - \max_{x_i\in\mathbf{x}_{1:t-1}} R_{\mathbf{x}_{1:t-1}}(x_i) \\
&\geq \sqrt{|\mathbf{x}_{1:t-1}|} \max_{x_i\in\mathbf{x}_{1:t-1}\cup\{x\}} R_{\mathbf{x}_{1:t-1}\cup\{x\}}(x_i) \\
&\geq \sqrt{|\mathbf{x}_{1:t-1}|} R_{\mathbf{x}_{1:t-1}\cup\{x\}}(x).
\end{aligned}$$

Since $\|K_{x\mathbf{x}_{1:t-1}}\| = \sqrt{\sum_{x_i\in\mathbf{x}_{1:t-1}} k_{x_i x}^2} \leq \sum_{x_i\in\mathbf{x}_{1:t-1}} k_{x_i x} = R_{\mathbf{x}_{1:t-1}\cup\{x\}}(x)$, it follows that $\psi_{min}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}) \geq \sqrt{|\mathbf{x}_{1:t-1}|}\|K_{x\mathbf{x}_{1:t-1}}\|$. Finally,

$$\begin{aligned}
&\|(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\|_2 \\
&= 1/(\psi_{min}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}) + \sigma_n^2 I) \\
&\leq 1/(\psi_{min}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}})) \\
&\leq 1/\big(\sqrt{|\mathbf{x}_{1:t-1}|}\|K_{x\mathbf{x}_{1:t-1}}\|\big).
\end{aligned}$$

□

Lemma A.10. *In the notations of Section A.5, if for all $t = 1, \ldots, T$ matrix $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$ is diagonally dominant (Definition 3.9), then $\|K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}\|_2 \leq 2\sigma_y^2$.*

Proof. Fix all $t = 1, \ldots, T$. By applying Gershgorin circle theorem to matrix $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$, for some point $x_i \in \mathbf{x}_{1:t-1}$:

$$\begin{aligned}
&|\psi_{max}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}) - k_{x_i x_i}| \\
&\leq \sum_{x_j\in\mathbf{x}_{1:t-1}\setminus x_i} k_{x_i x_j} \\
&\leq k_{x_i x_i}/\big(\sqrt{|\mathbf{x}_{1:t-1}|-1}+1\big) \\
&= \sigma_y^2/\big(\sqrt{|\mathbf{x}_{1:t-1}|-1}+1\big)
\end{aligned}$$

where $\psi_{max}(\cdot)$ denotes the largest eigenvalue of a matrix, the second inequality is due to diagonal dominance property of matrix $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$ and the equality is due to $k_{x_i x_i} = \sigma_y^2$ for every x_i. Since $K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}$ is a symmetric, positive-semidefinite

matrix, it follows that

$$\begin{aligned}
&\|K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}\|_2 \\
&= \psi_{max}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}}) \\
&\leq \sigma_y^2/\big(\sqrt{|\mathbf{x}_{1:t-1}|-1}+1\big) + k_{x_i x_i} \\
&\leq \sigma_y^2\big(1+1/(\sqrt{|\mathbf{x}_{1:t-1}|-1}+1)\big) \\
&\leq 2\sigma_y^2.
\end{aligned}$$

. □

Lemma A.11. *In the notations of Section A.5, for all $t = 1, \ldots, T$ and any unobserved input $x \in \mathcal{X}$ at iteration t*

$$\|K_{x\mathbf{x}_{1:t-1}}\|^2 \cdot \psi_{min}((K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}) \leq K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{x}_{1:t-1}x}$$

where $\psi_{min}(\cdot)$ denotes the smallest eigenvalue of a matrix.

Proof. Since $(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}$ is a symmetric, positive-definite matrix, there exists an orthonormal basis comprising the eigenvectors $E \triangleq [e_1 \ldots e_{|\mathbf{x}_{1:t-1}|}]$ ($e_i^\top e_i = 1$ and $e_i^\top e_j = 0$ for $i \neq j$) and their associated positive eigenvalues $\Psi^{-1} \triangleq \text{Diag}[\psi_1^{-1}, \ldots, \psi_{|\mathbf{x}_{1:t-1}|}^{-1}]$ such that $(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} = E\Psi^{-1}E^\top$ (i.e., spectral theorem). Denote $\{p_i\}_{i=1}^{|\mathbf{x}_{1:t-1}|}$ as the set of coefficients when $K_{\mathbf{x}_{1:t-1}x}$ is projected on E. Then

$$\begin{aligned}
&K_{x\mathbf{x}_{1:t-1}}(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} K_{\mathbf{x}_{1:t-1}x} \\
&= \left(\sum_{i=1}^{|\mathbf{x}_{1:t-1}|} p_i e_i^\top\right)(K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}\left(\sum_{i=1}^{|\mathbf{x}_{1:t-1}|} p_i e_i\right) \\
&= \left(\sum_{i=1}^{|\mathbf{x}_{1:t-1}|} p_i e_i^\top\right)\left(\sum_{i=1}^{|\mathbf{x}_{1:t-1}|} p_i (K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1} e_i\right) \\
&= \left(\sum_{i=1}^{|\mathbf{x}_{1:t-1}|} p_i e_i^\top\right)\left(\sum_{i=1}^{|\mathbf{x}_{1:t-1}|} p_i \psi_i^{-1} e_i\right) \\
&= \sum_{i=1}^{|\mathbf{x}_{1:t-1}|} p_i^2 \psi_i^{-1} \\
&\geq \psi_{min}((K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1}) \sum_{i=1}^{|\mathbf{x}_{1:t-1}|} p_i^2 \\
&= \psi_{min}((K_{\mathbf{x}_{1:t-1}\mathbf{x}_{1:t-1}} + \sigma_n^2 I)^{-1})\, \|K_{x\mathbf{x}_{1:t-1}}\|^2.
\end{aligned}$$

□

Appendix B

Appendix for Chapter 4

B.1 Proofs and Derivations

B.1.1 Derivation of (4.3)

The second summand on RHS of (4.2) can be re-written as

$$\begin{aligned}
&\mathbb{I}[f(\mathcal{X}); \mathbf{y}_{1:H}|d_0, \pi] \\
&= \sum_{t=1}^{H} [f(x)_{x\in\mathcal{X}}; \mathbf{y}_t|d_{t-1}, \pi] \\
&= 0.5 \sum_{t=0}^{H-1} \log |I + \sigma_n^{-2}\Sigma_t^{\pi}(\mathbf{x}_{t+1})| \ .
\end{aligned} \tag{B.1}$$

The first equality is due to the chain rule for mutual information [Cover and Thomas, 2006]. Let $\mathbf{x}_{t-1} \triangleq (x_{t-1,1}, \ldots, x_{t-1,\kappa,})$. The last equality follows from

$$\begin{aligned}
&\mathbb{I}[f(\mathcal{X}); \mathbf{y}_t|d_{t-1}, \pi] \\
&= \mathbb{H}[\mathbf{y}_t|d_{t-1}, \pi] - \mathbb{H}[\mathbf{y}_t|d_{t-1}, f(x)_{x\in\mathcal{X}}, \pi] \\
&= \mathbb{H}[\mathbf{y}_t|d_{t-1}, \pi] - \mathbb{H}[\mathbf{y}_t|f(x_{t-1,1}), \ldots, f(x_{t-1,\kappa}), \pi] \\
&= 0.5\kappa \log(2\pi e) + 0.5 \log |\sigma_n^2 I + \Sigma_{t-1}^{\pi}(\mathbf{x}_t)| - 0.5\kappa \log(2\pi e) - 0.5 \log |\sigma_n^2 I| \\
&= 0.5 \log(|\sigma_n^2 I + \Sigma_{t-1}^{\pi}(\mathbf{x}_t)| \ |\sigma_n^2 I|^{-1}) \\
&= 0.5 \log(|\sigma_n^2 I + \Sigma_{t-1}^{\pi}(\mathbf{x}_t)||\sigma_n^{-2} I|) \\
&= 0.5 \log |I + \sigma_n^{-2}\Sigma_{t-1}^{\pi}(\mathbf{x}_t)|
\end{aligned} \tag{B.2}$$

where the first equality is due to the definition of conditional mutual information, the third equality is due to the definition of Gaussian entropy, that is, $\mathbb{H}[\mathbf{y}_t|d_{t-1}, \pi] \triangleq 0.5\kappa \log(2\pi e) + 0.5 \log |\sigma_n^2 I + \Sigma_{t-1}^{\pi}(\mathbf{x}_t)|$ and $\mathbb{H}[\mathbf{y}_t|f(x_{t-1,1}), \ldots, f(x_{t-1,\kappa}), \pi] \triangleq 0.5\kappa \log(2\pi e) +$

$0.5 \log |\sigma_n^2 I|$, the latter of which follows from $\varepsilon = y_{t,i} - f x_{t,i} \sim \mathcal{N}(0, \sigma_n^2)$ for stage $t = 0, \ldots, H-1$ and $i = 1, \ldots, \kappa$, and hence $p(\mathbf{y}_t | f(x_{t-1,1}), \ldots, f(x_{t-1,\kappa}), \pi) = \mathcal{N}(\mathbf{0}, \sigma_n^2 I)$. So, (4.2) can be re-expressed as

$$V_0^\pi(d_0) = \mathbb{E}_{\mathbf{y}_{1:H}|d_0,\pi}[\mathbf{1}^\top \mathbf{y}_{1:H}] + 0.5 \sum_{t=0}^{H-1} \log |I + \sigma_n^{-2} \Sigma_t^\pi(\mathbf{x}_{t+1})| \ . \tag{B.3}$$

Given an arbitrary positive integer $'$ and denoting $\mathbf{y}_{\tau+1:'}$ as a vector of output measurements from stage $\tau + 1$ to stage H', (B.3) for $H = 1, \ldots, H'$ are, respectively, equivalent to

$$V_\tau^\pi(d_\tau) = \mathbb{E}_{\mathbf{y}_{\tau+1:H'}|d_\tau,\pi}[\mathbf{1}^\top \mathbf{y}_{\tau+1:H'}] + 0.5 \sum_{t=0}^{H'-1} \log |I + \sigma_n^{-2} \Sigma_t^\pi(\mathbf{x}_{t+1})| \ . \tag{B.4}$$

for $\tau = H'-1, \ldots, 0$ by simply adding τ to the indices denoting the iteration in (B.3). From (B.4),

$$
\begin{aligned}
&V_\tau^\pi(d_\tau) \\
&= \mathbb{E}_{\mathbf{y}_{\tau+1:H'}|d_\tau,\pi}[\mathbf{1}^\top \mathbf{y}_{\tau+1:H'}] + 0.5\beta \sum_{t=\tau}^{H'-1} \log|I + \sigma_n^{-2}\Sigma_t^\pi(\mathbf{x}_{t+1})| \\
&= \int \mathbf{1}^\top \mathbf{y}_{\tau+1:H'}\ p(\mathbf{y}_{\tau+1:H'}|d_\tau,\pi)\ \mathrm{d}\mathbf{y}_{\tau+1:H'} + 0.5\beta \sum_{t=\tau}^{H'-1} \log|I + \sigma_n^{-2}\Sigma_t^\pi(\mathbf{x}_{t+1})| \\
&= \int (\mathbf{1}^\top \mathbf{y}_{\tau+1} + \mathbf{1}^\top \mathbf{y}_{\tau+2:H'})\ p(\mathbf{y}_{\tau+2:H'}|d_{\tau+1},\pi)\ \mathrm{d}\mathbf{y}_{\tau+2:H'}\ p(\mathbf{y}_{\tau+1}|d_\tau,\pi)\ \mathrm{d}\mathbf{y}_{\tau+1} \\
&\quad + 0.5\beta \sum_{t=\tau}^{H'-1} \log|I + \sigma_n^{-2}\Sigma_t^\pi(\mathbf{x}_{t+1})| \\
&= \int \mathbf{1}^\top \mathbf{y}_{\tau+1} \int p(\mathbf{y}_{\tau+2:H'}|d_{\tau+1},\pi)\ \mathrm{d}\mathbf{y}_{\tau+2:H'}\ p(\mathbf{y}_{\tau+1}|d_\tau,\pi)\ \mathrm{d}\mathbf{y}_{\tau+1} \\
&\quad + 0.5\beta\ \log|I + \sigma_n^{-2}\Sigma_t^\pi(\mathbf{x}_{t+1})| \\
&\quad + \int \mathbf{1}^\top \mathbf{y}_{\tau+2:H'}\ p(\mathbf{y}_{\tau+2:H'}|d_{\tau+1},\pi)\ \mathrm{d}\mathbf{y}_{\tau+2:H'}\ p(\mathbf{y}_{\tau+1}|d_\tau,\pi)\ \mathrm{d}\mathbf{y}_{\tau+1} \\
&\quad + 0.5\beta \sum_{t=\tau+1}^{H'-1} \log|I + \sigma_n^{-2}\Sigma_t^\pi(\mathbf{x}_{t+1})| \\
&= \int \mathbf{1}^\top \mathbf{y}_{\tau+1}\ p(\mathbf{y}_{\tau+1}|d_\tau,\pi)\ \mathrm{d}\mathbf{y}_{\tau+1} + \ 0.5\beta\ \log|I + \sigma_n^{-2}\Sigma_t^\pi(\mathbf{x}_{t+1})| \\
&\quad + \int\int \mathbf{1}^\top \mathbf{y}_{\tau+2:H'}\ p(\mathbf{y}_{\tau+2:H'}|d_{\tau+1},\pi)\ \mathrm{d}\mathbf{y}_{\tau+2:H'} \\
&\quad + 0.5\beta \sum_{t=\tau+1}^{H'-1} \log|I + \sigma_n^{-2}\Sigma_t^\pi(\mathbf{x}_{t+1})|\ p(\mathbf{y}_{\tau+1}|d_\tau,\pi)\ \mathrm{d}\mathbf{y}_{\tau+1} \\
&= \mathbf{1}^\top \mu_\tau^\pi(\mathbf{x}_{\tau+1}) + \ 0.5\beta\ \log|I + \sigma_n^{-2}\Sigma_\tau^\pi(\mathbf{x}_{\tau+1})| \\
&\quad + \int \mathbb{E}_{\mathbf{y}_{\tau+2:H'}|d_{\tau+1},\pi}[\mathbf{1}^\top \mathbf{y}_{\tau+2:H'}] + 0.5\beta \sum_{t=\tau+1}^{H'-1} \log|I + \sigma_n^{-2}\Sigma_t^\pi(\mathbf{x}_{t+1})|\ p(\mathbf{y}_{\tau+1}|d_\tau,\pi)\ \mathrm{d}\mathbf{y}_{\tau+1} \\
&= \mathbf{1}^\top \mu_\tau^\pi(\pi(d_\tau)) + \ 0.5\beta\ \log|I + \sigma_n^{-2}\Sigma_t^\pi(\pi(d_\tau))| + \int V_{\tau+1}^\pi(d_{\tau+1})\ p(\mathbf{y}_{\tau+1}|d_\tau,\pi)\ \mathrm{d}\mathbf{y}_{\tau+1} \\
&= R(\pi(d_\tau), d_\tau) + \mathbb{E}_{\mathbf{y}_{t+1}|\pi(d_\tau),d_\tau}[V_{\tau+1}^\pi(\langle \mathbf{x}_{1:t} \oplus \pi(d_\tau), \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1}\rangle)] \\
&= Q_\tau^\pi(\pi(d_\tau), d_\tau)
\end{aligned}
$$

for stages $\tau = 0, \ldots, H'-1$ where the third last equality is due to (B.4) and the last two equalities follow from the definitions of R and Q_τ^π in (4.4) and (4.3), respectively. Note that in order to avoid the overloaded notations, we omit the upper index π from Σ_t^π and μ_t^π.

B.1.2 Lipschitz Continuity of $R(\mathbf{x}_{t+1}, d_t)$ (4.4)

Lemma B.1. *Let*

$$\alpha(\mathbf{x}_{1:t+1}) \triangleq \|K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}(K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} + \sigma_n^2 I)^{-1}\|_F$$

where matrices K are defined in (4.1), $\|\cdot\|_F$ is the Frobenius norm of a matrix and $d'_t \triangleq \langle \mathbf{x}_{1:t}, \mathbf{y}'_{1:t} \rangle$. Then,

$$|R(\mathbf{x}_{t+1}, d_t) - R(\mathbf{x}_{t+1}, d'_t)| \leq \sqrt{\kappa}\ \alpha(\mathbf{x}_{1:t+1}) \|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\|.$$

—

Proof.

$$\begin{aligned}
&|R(\mathbf{x}_{t+1}, d_t) - R(\mathbf{x}_{t+1}, d'_t)| \\
&= |\mathbf{1}^\top(\mu_{t|d_t}(\mathbf{x}_{t+1}) - \mu_{t|d'_t}(\mathbf{x}_{t+1}))| \\
&\leq \|\mu_{t|d_t}(\mathbf{x}_{t+1}) - \mu_{t|d'_t}(\mathbf{x}_{t+1})\|_1 \\
&= \|K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}(K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} + \sigma_n^2 I)^{-1}(\mathbf{y}_{1:t} - \mathbf{y}'_{1:t})^\top\|_1 \\
&\leq \sqrt{\kappa}\ \|K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}(K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} + \sigma_n^2 I)^{-1}(\mathbf{y}_{1:t} - \mathbf{y}'_{1:t})^\top\| \\
&= \sqrt{\kappa}\ \|K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}(K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} + \sigma_n^2 I)^{-1}(\mathbf{y}_{1:t} - \mathbf{y}'_{1:t})^\top\|_F \\
&\leq \sqrt{\kappa}\ \|K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}(K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} + \sigma_n^2 I)^{-1}\|_F \|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\|_F \\
&= \sqrt{\kappa}\ \|K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}(K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} + \sigma_n^2 I)^{-1}\|_F \|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\| \\
&= \sqrt{\kappa}\ \alpha(\mathbf{x}_{1:t+1}) \|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\|\ .
\end{aligned}$$

Conditioning in posterior means $\mu_{t|d_t}(\mathbf{x}_{t+1})$ and $\mu_{t|d'_t}(\mathbf{x}_{t+1})$ reflects the fact that they are computed using d_t and d'_t, respectively. The first equality is due to (4.4). The first inequality is due to triangle inequality. The second equality is due to (4.1). The second inequality follows from a property of vector norms (see Section 2.2.2 in [Golub and Van Loan, 1996]). The last inequality is due to the submultiplicativity of the Frobenius norm (see Section II.2.1 in [Stewart and Sun, 1990]). The last equality follows from the definition of $\alpha(\mathbf{x}_{1:t+1})$. □

B.1.3 Lipschitz Continuity of $V_t^*(d_t)$ (4.5)

Definition B.1. *Let $L_H(\mathbf{x}_{1:H}) \triangleq 0$. Define*

$$L_t(\mathbf{x}_{1:t}) \triangleq \max_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} \sqrt{\kappa}\ \alpha(\mathbf{x}_{1:t+1}) + L_{t+1}(\mathbf{x}_{1:t+1}) \sqrt{1 + \alpha(\mathbf{x}_{1:t+1})^2}$$

for $t = 0, \ldots, H-1$ where the function α is previously defined in Lemma B.1.

The following result shows that $V_t^*(d_t)$ (4.5) is Lipschitz continuous in the output measurements $\mathbf{y}_{1:t}$ with Lipschitz constant $L_t(\mathbf{x}_{1:t})$:

Theorem B.1. *For $t = 0, \ldots, H$,*

$$|V_t^*(d_t) - V_t^*(d'_t)| \leq L_t(\mathbf{x}_{1:t})\|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\| \tag{B.5}$$

where d'_t is previously defined in Lemma B.1.

Proof. We give a proof by induction on t. When $t = H$ (i.e., base case), $V_H^*(d_H) = 0$ for any d_H. So, $|V_H^*(d_H) - V_H^*(d'_H)| = 0 \leq L_H(\mathbf{x}_{1:H})\|\mathbf{y}_{1:H} - \mathbf{y}'_{1:H}\|$. Supposing (B.5) holds for $t+1$ (i.e., induction hypothesis), we will prove that it holds for $t = 0, \ldots, H-1$. Let $\mathbf{x}_{t+1}^* \triangleq \pi^*(d_t)$ and $\Delta_{t+1} \triangleq \mu_{t|d_t}(\mathbf{x}_{t+1}^*) - \mu_{t|d'_t}(\mathbf{x}_{t+1}^*)|$. Using (4.1), the submultiplicativity of the Frobenius norm (see Section II.2.1 in [Stewart and Sun, 1990]), and the definition of $\alpha(\mathbf{x}_{1:t+1})$,

$$\|\Delta_{t+1}\| \leq \alpha(\mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1}^*)\|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\| \ . \tag{B.6}$$

Without loss of generality, assume that $V_t^*(d_t) \geq V_t^*(d'_t)$. From (4.5),

$$\begin{aligned}
&V_t^*(d_t) - V_t^*(d'_t) \\
&\leq Q_t^*(\mathbf{x}_{t+1}^*, d_t) - Q_t^*(\mathbf{x}_{t+1}^*, d'_t) \\
&\leq |Q_t^*(\mathbf{x}_{t+1}^*, d_t) - Q_t^*(\mathbf{x}_{t+1}^*, d'_t)| \\
&\leq \left|R(\mathbf{x}_{t+1}^*, d_t) - R(\mathbf{x}_{t+1}^*, d'_t)\right| + \Bigg| \int p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^*, d_t)\ V_{t+1}^*(\langle \mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1}^*, \mathbf{y}_{1:t+1}\rangle)\ \mathrm{d}\mathbf{y}_{t+1} \\
&\quad - \int p(\mathbf{y}'_{t+1}|\mathbf{x}_{t+1}^*, d'_t)\ V_{t+1}^*(\langle \mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1}^*, \mathbf{y}'_{1:t+1}\rangle)\ \mathrm{d}\mathbf{y}'_{t+1}\Bigg| \\
&\leq \sqrt{\kappa}\ \alpha(\mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1}^*)\|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\| + \int p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^*, d_t)\ L_{t+1}(\mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1}^*)\|(\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}) \oplus \Delta_{t+1}\|\ \mathrm{d}\mathbf{y}_{t+1} \\
&= \sqrt{\kappa}\ \alpha(\mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1}^*)\|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\| + L_{t+1}(\mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1}^*)\|(\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}) \oplus \Delta_{t+1}\| \\
&\leq \sqrt{\kappa}\ \alpha(\mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1}^*)\|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\| + L_{t+1}(\mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1}^*)\sqrt{1 + \alpha(\mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1}^*)^2}\ \|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\| \\
&\leq L_t(\mathbf{x}_{1:t})\|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\|
\end{aligned} \tag{B.7}$$

where the third inequality follows from (4.5) and triangle inequality, the fourth inequality follows from Lemma B.1, change of variable $\mathbf{y}'_{t+1} \triangleq \mathbf{y}_{t+1} - \Delta_{t+1}$, and the induction hypothesis, the second last inequality in (B.7) is due to

$$\|(\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}) \oplus \Delta_{t+1}\| = \sqrt{\|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\|^2 + \|\Delta_{t+1}\|^2} \leq \sqrt{1 + \alpha(\mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1}^*)^2}\ \|\mathbf{y}_{1:t} - \mathbf{y}'_{1:t}\|$$

with the inequality following from (B.6), and the last inequality in (B.7) is due to the definition of L_t (Definition B.1). □

B.1.4 Approximation Quality of $\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ (4.6)

There are two sources of error arising in using $\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ to approximate $Q_t^*(\mathbf{x}_{t+1}, d_t)$: (a) Every stage-wise expectation term in (4.5) is approximated via stochastic sampling (4.6) of a finite number N of i.i.d. multivariate Gaussian vectors $\mathbf{y}^1, \ldots, \mathbf{y}^N$ from the GP posterior belief $p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, d_t) = \mathcal{N}(\mu_t(\mathbf{x}_{t+1}), \Sigma_t(\mathbf{x}_{t+1}))$ (4.1) and (b) evaluating $\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ does not involve utilizing the values of V_{t+1}^* but rather that of its approximation $\mathcal{V}_{t+1}$. To facilitate capturing the error due to finite stochastic sampling described in (a), the following intermediate function is introduced:

$$\mathcal{U}_t(\mathbf{x}_{t+1}, d_t) \triangleq R(\mathbf{x}_{t+1}, d_t) + \frac{1}{N}\sum_{\ell=1}^{N} V_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) \tag{B.8}$$

for $t = 0, \ldots, H-1$. The following lemma shows that $\mathcal{U}_t(\mathbf{x}_{t+1}, d_t)$ can approximate $Q_t^*(\mathbf{x}_{t+1}, d_t)$ arbitrarily closely:

Lemma B.2. *Suppose that the observations $d_{t'}$, $H \in \mathbb{Z}^+$, a budget of $\kappa(H-t')$ input locations for $t' = 0, \ldots, H-1$, $\lambda > 0$, and $N \in \mathbb{Z}^+$ are given. For all tuples $\langle t, \mathbf{x}_{t+1}, d_t \rangle$ generated at stage $t = t', \ldots, H-1$ by (4.6) to compute $\mathcal{V}_{t'}(d_{t'})$,*

$$P(|\mathcal{U}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \leq \lambda) \geq 1 - 2\exp\left(-\frac{N\lambda^2}{2K^2}\right)$$

where $K \triangleq \mathcal{O}(\kappa^H \sqrt{H!}\ \sigma_n(1+\sigma_y^2/\sigma_n^2)^H)$.

Proof. For any tuple $\langle t, \mathbf{x}_{t+1}, d_t \rangle$, define the following auxiliary function:

$$\begin{aligned} \mathcal{G}(\mathbf{y}^1, \ldots, \mathbf{y}^N) &\triangleq \frac{1}{N}\sum_{\ell=1}^{N} V_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) \\ &= \mathcal{U}_t(\mathbf{x}_{t+1}, d_t) - R(\mathbf{x}_{t+1}, d_t) \end{aligned} \tag{B.9}$$

which follows from (B.8). Taking an expectation of (B.9) with respect to GP posterior

belief $p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, d_t) = \mathcal{N}(\mu_t(\mathbf{x}_{t+1}), \Sigma_t(\mathbf{x}_{t+1}))$ gives

$$\begin{aligned}
&\mathbb{E}_{\mathbf{y}^1,\dots,\mathbf{y}^N \sim \mathcal{N}(\mu_t(\mathbf{x}_{t+1}),\Sigma_t(\mathbf{x}_{t+1}))}\left[\mathcal{G}(\mathbf{y}^1,\dots,\mathbf{y}^N)\right] \\
&= \mathbb{E}_{\mathbf{y}^1,\dots,\mathbf{y}^N \sim \mathcal{N}(\mu_t(\mathbf{x}_{t+1}),\Sigma_t(\mathbf{x}_{t+1}))}\left[\frac{1}{N}\sum_{\ell=1}^{N} V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle)\right] \\
&= \frac{1}{N}\sum_{\ell=1}^{N} \mathbb{E}_{\mathbf{y}^1,\dots,\mathbf{y}^N \sim \mathcal{N}(\mu_t(\mathbf{x}_{t+1}),\Sigma_t(\mathbf{x}_{t+1}))}[V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle)] \\
&= \frac{1}{N}\sum_{\ell=1}^{N} \mathbb{E}_{\mathbf{y}^\ell \sim \mathcal{N}(\mu_t(\mathbf{x}_{t+1}),\Sigma_t(\mathbf{x}_{t+1}))}[V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle)] \\
&= \frac{1}{N}\sum_{\ell=1}^{N} \mathbb{E}_{\mathbf{y}_{t+1}|\mathbf{x}_{t+1},d_t}[V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle)] \\
&= \mathbb{E}_{\mathbf{y}_{t+1}|\mathbf{x}_{t+1},d_t}[V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle)] \\
&= Q^*_t(\mathbf{x}_{t+1}, d_t) - R(\mathbf{x}_{t+1}, d_t)
\end{aligned} \tag{B.10}$$

such that the last equality is due to (4.5). From (B.9) and (B.10),

$$|\mathcal{U}_t(\mathbf{x}_{t+1}, d_t) - Q^*_t(\mathbf{x}_{t+1}, d_t)| = \left|\mathcal{G}(\mathbf{y}^1,\dots,\mathbf{y}^N) - \mathbb{E}_{\mathbf{y}^1,\dots,\mathbf{y}^N \sim \mathcal{N}(\mu_t(\mathbf{x}_{t+1}),\Sigma_t(\mathbf{x}_{t+1}))}\left[\mathcal{G}(\mathbf{y}^1,\dots,\mathbf{y}^N)\right]\right| . \tag{B.11}$$

The RHS of (B.11) can usually be bounded using a concentration inequality that involves independent Gaussian random variables. However, the components of the multivariate Gaussian vector $\mathbf{y}^\ell$ are correlated. To resolve this complication, we exploit a change of variables trick to make the components independent:

$$\mathbf{y}^\ell = \mu_t(\mathbf{x}_{t+1}) + \Psi \mathbf{z}^\ell \tag{B.12}$$

for $\ell = 1, \dots, N$ where Ψ is a $\kappa \times \kappa$ lower triangular matrix satisfying the Cholesky decomposition of the symmetric and positive definite $\Sigma_t(\mathbf{x}_{t+1}) = \Psi\Psi^\top$ and $\mathbf{z}^\ell$ is a standard multivariate Gaussian vector with independent components (see Section 53.2.2 in [Taboga, 2017]).

Define a new auxiliary function G in terms of $\mathcal{G}$ by plugging (B.12) into (B.9):

$$G(\mathbf{z}^1,\dots,\mathbf{z}^N) \triangleq \mathcal{G}(\mathbf{y}^1,\dots,\mathbf{y}^N) . \tag{B.13}$$

We will first prove that G is Lipschitz continuous in $\mathbf{z}^1 \oplus \dots \oplus \mathbf{z}^N$ with Lipschitz constant $L_{t+1}(\mathbf{x}_{1:t+1})\sqrt{\mathrm{Tr}(\Sigma_t(\mathbf{x}_{t+1}))/N}$, which is a sufficient condition for using the Tsirelson-Ibragimov-Sudakov inequality [Boucheron *et al.*, 2013] to prove the probabilistic bound in Lemma B.2. To simplify notations, let $\overline{\mathbf{z}} \triangleq \mathbf{z}^1 \oplus \dots \oplus \mathbf{z}^N$ and

$\overline{\mathbf{z}}' \triangleq \mathbf{z}'^1 \oplus \ldots \oplus \mathbf{z}'^N$. Then,

$$
\begin{aligned}
&|G(\mathbf{z}^1, \ldots, \mathbf{z}^N) - G(\mathbf{z}'^1, \ldots, \mathbf{z}'^N)| \\
&= |\mathcal{G}(\mathbf{y}^1, \ldots, \mathbf{y}^N) - \mathcal{G}(\mathbf{y}'^1, \ldots, \mathbf{y}'^N)| \\
&\leq \frac{1}{N} \sum_{\ell=1}^{N} \left| V_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) - V_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}'^\ell \rangle) \right| \\
&\leq \frac{L_{t+1}(\mathbf{x}_{1:t+1})}{N} \sum_{\ell=1}^{N} \|\mathbf{y}^\ell - \mathbf{y}'^\ell\| \\
&\leq \frac{L_{t+1}(\mathbf{x}_{1:t+1})}{N} \sqrt{N} \|\Psi\|_F \|\overline{\mathbf{z}} - \overline{\mathbf{z}}'\| \\
&= \frac{L_{t+1}(\mathbf{x}_{1:t+1})}{\sqrt{N}} \|\Psi\|_F \|\overline{\mathbf{z}} - \overline{\mathbf{z}}'\| \\
&= L_{t+1}(\mathbf{x}_{1:t+1}) \sqrt{\frac{\mathrm{Tr}(\Sigma_t(\mathbf{x}_{t+1}))}{N}} \|\overline{\mathbf{z}} - \overline{\mathbf{z}}'\|
\end{aligned}
\tag{B.14}
$$

where the first equality is due to (B.13), the last equality follows from a property of Frobenius norm (see Section 10.4.3 in [Petersen and Pedersen, 2012]), the first inequality is due to (B.9) and triangle inequality, the second inequality is a direct consequence of Theorem B.1 in Appendix B.1.3, and the third inequality follows from

$$
\begin{aligned}
&\sum_{\ell=1}^{N} \|\mathbf{y}^\ell - \mathbf{y}'^\ell\| \\
&= \sum_{\ell=1}^{N} \|\Psi(\mathbf{z}^\ell - \mathbf{z}'^\ell)\| \\
&= \sum_{\ell=1}^{N} \|\Psi(\mathbf{z}^\ell - \mathbf{z}'^\ell)\|_F \\
&\leq \sum_{\ell=1}^{N} \|\Psi\|_F \|\mathbf{z}^\ell - \mathbf{z}'^\ell\|_F \\
&= \|\Psi\|_F \sum_{\ell=1}^{N} \|\mathbf{z}^\ell - \mathbf{z}'^\ell\| \\
&\leq \sqrt{N} \|\Psi\|_F \|\overline{\mathbf{z}} - \overline{\mathbf{z}}'\|
\end{aligned}
$$

where the first equality is due to (B.12), the first inequality is due to the submultiplicativity of the Frobenius norm (see Section II.2.1 in [Stewart and Sun, 1990]), and the last inequality is due to Cauchy-Schwarz inequality. Since conditioning does not

increase GP posterior variance,

$$\mathrm{Tr}(\Sigma_t(\mathbf{x}_{t+1})) \leq \mathrm{Tr}(K_{\mathbf{x}_{t+1}\mathbf{x}_{t+1}}) = \kappa(\sigma_y^2 + \sigma_n^2) \ . \tag{B.15}$$

From (B.15) and Lemma B.9,

$$\begin{aligned} & L_{t+1}(\mathbf{x}_{1:t+1})\sqrt{\mathrm{Tr}(\Sigma_t(\mathbf{x}_{t+1}))} \\ &= \mathcal{O}(\kappa^{H-t-1/2}\sqrt{H!/(t+1)!}\ (1+\sigma_y^2/\sigma_n^2)^{H-t-1})\ \mathcal{O}(\kappa^{1/2}(\sigma_y^2+\sigma_n^2)^{1/2}) \\ &= \mathcal{O}(\kappa^{H-t}\sqrt{H!/(t+1)!}\ \sigma_n(1+\sigma_y^2/\sigma_n^2)^{H-t-1/2}) \ . \end{aligned} \tag{B.16}$$

It follows from (B.16) that

$$K \triangleq \max_{\langle t, \mathbf{x}_{t+1}, d_t\rangle} L_{t+1}(\mathbf{x}_{1:t+1})\sqrt{\mathrm{Tr}(\Sigma_t(\mathbf{x}_{t+1}))} = \mathcal{O}(\kappa^H\sqrt{H!}\ \sigma_n(1+\sigma_y^2/\sigma_n^2)^H) \ . \tag{B.17}$$

Finally,

$$\begin{aligned} & P(|\mathcal{U}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| > \lambda) \\ &= P(|\mathcal{G}(\mathbf{y}^1, \ldots, \mathbf{y}^N) - \mathbb{E}_{\mathbf{y}^1,\ldots,\mathbf{y}^N}[\mathcal{G}(\mathbf{y}^1, \ldots, \mathbf{y}^N)]| > \lambda) \\ &= P(|G(\mathbf{z}^1, \ldots, \mathbf{z}^N) - \mathbb{E}_{\mathbf{z}^1,\ldots,\mathbf{z}^N}[G(\mathbf{z}^1, \ldots, \mathbf{z}^N)]| > \lambda) \\ &\leq 2\exp\left(-\frac{N\lambda^2}{2L_{t+1}^2(\mathbf{x}_{1:t+1})\mathrm{Tr}(\Sigma_t(\mathbf{x}_{t+1}))}\right) \\ &\leq 2\exp\left(-\frac{N\lambda^2}{2K^2}\right) \end{aligned}$$

where the first equality is due to (B.11), the second equality is due to (B.13) above and (B.18) below, the first inequality is due to the Tsirelson-Ibragimov-Sudakov inequality that requires G to be Lipschitz continuous in $\mathbf{z}^1 \oplus \ldots \oplus \mathbf{z}^N$ which is shown in (B.14) (see Section 5.4 on page 125 in [Boucheron *et al.*, 2013]), and the last

inequality is due to (B.17).

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{y}^1,\dots,\mathbf{y}^N}[\mathcal{G}(\mathbf{y}^1,\dots,\mathbf{y}^N)] \\
&= \mathbb{E}_{\mathbf{y}_{t+1}|\mathbf{x}_{t+1},d_t}[V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1}\rangle)] \\
&= \int_{\mathbb{R}^\kappa} V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1}\rangle)\; p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, d_t)\; \mathrm{d}\mathbf{y}_{t+1} \\
&= \int_{\mathbb{R}^\kappa} V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus (\mu_t(\mathbf{x}_{t+1}) + \Psi\mathbf{z}_{t+1})\rangle) \frac{1}{|\Psi|} p(\mathbf{z}_{t+1}) \left|\frac{\partial \mathbf{y}_{t+1}}{\partial \mathbf{z}_{t+1}}\right| \mathrm{d}\mathbf{z}_{t+1} \\
&= \int_{\mathbb{R}^\kappa} V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus (\mu_t(\mathbf{x}_{t+1}) + \Psi\mathbf{z}_{t+1})\rangle)\; p(\mathbf{z}_{t+1})\; \mathrm{d}\mathbf{z}_{t+1} \\
&= \mathbb{E}_{\mathbf{z}_{t+1}}[V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus (\mu_t(\mathbf{x}_{t+1}) + \Psi\mathbf{z}_{t+1})\rangle)] \\
&= \mathbb{E}_{\mathbf{z}^1,\dots,\mathbf{z}^N}\left[\frac{1}{N}\sum_{\ell=1}^{N} V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus (\mu_t(\mathbf{x}_{t+1}) + \Psi\mathbf{z}^\ell)\rangle)\right] \\
&= \mathbb{E}_{\mathbf{z}^1,\dots,\mathbf{z}^N}[G(\mathbf{z}^1,\dots,\mathbf{z}^N)]
\end{aligned}
\tag{B.18}
$$

where the first equality is due to (B.10), the third equality follows from (B.12), $p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, d_t) = p(\mathbf{z}_{t+1} = \Psi^{-1}(\mathbf{y}_{t+1} - \mu_t(\mathbf{x}_{t+1})))/|\Psi|$ (see Section 35.1.2 in [Taboga, 2017]), and an integration by substitution for multiple variables, the fourth equality is due to $|\partial \mathbf{y}_{t+1}/\partial \mathbf{z}_{t+1}| = |\Psi|$, and the last two equalities can be derived in a similar manner as (B.10) using (B.13). □

Lemma B.3. *Suppose that the observations $d_{t'}$, $H \in \mathbb{Z}^+$, a budget of $\kappa(H - t')$ input locations for $t' = 0, \dots, H-1$, $\lambda > 0$, and $N \in \mathbb{Z}^+$ are given. The probability of $|\mathcal{U}_t(\mathbf{x}_{t+1}, d_t) - Q^*_t(\mathbf{x}_{t+1}, d_t)| \leq \lambda$ for all tuples $\langle t, \mathbf{x}_{t+1}, d_t\rangle$ generated at stage $t = t', \dots, H-1$ by (4.6) to compute $\mathcal{V}_{t'}(d_{t'})$ is at least*

$$1 - 2\,(NA)^H \exp\left(-\frac{N\lambda^2}{2K^2}\right)$$

where K is previously defined in Lemma B.2.

Proof. From Lemma B.2,

$$P(|\mathcal{U}_t(\mathbf{x}_{t+1}, d_t) - Q^*_t(\mathbf{x}_{t+1}, d_t)| > \lambda) \leq 2\exp\left(-\frac{N\lambda^2}{2K^2}\right)$$

for each tuple $\langle t, \mathbf{x}_{t+1}, d_t\rangle$ generated at stage $t = t', \dots, H-1$ by (4.6) to compute $\mathcal{V}_{t'}(d_{t'})$. Since there will be no more than $(NA)^H$ tuples $\langle t, \mathbf{x}_{t+1}, d_t\rangle$ generated at stage $t = t', \dots, H-1$ by (4.6) to compute $\mathcal{V}_{t'}(d_{t'})$, the probability of $|\mathcal{U}_t(\mathbf{x}_{t+1}, d_t) - Q^*_t(\mathbf{x}_{t+1}, d_t)| > \lambda$ for some generated tuple $\langle t, \mathbf{x}_{t+1}, d_t\rangle$ is at most $2(NA)^H \exp(-N\lambda^2/(2K^2))$ by applying the union bound. Lemma B.3 directly follows. □

Lemma B.4. *Suppose that the observations $d_{t'}$, $H \in \mathbb{Z}^+$, a budget of $\kappa(H-t')$ input locations for $t' = 0, \ldots, H-1$, $\lambda > 0$, and $N \in \mathbb{Z}^+$ are given. If*

$$|\mathcal{U}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \leq \lambda \tag{B.19}$$

for all tuples $\langle t, \mathbf{x}_{t+1}, d_t \rangle$ generated at stage $t = t', \ldots, H-1$ by (4.6) to compute $\mathcal{V}_{t'}(d_{t'})$, then, for all $\mathbf{x}_{t'+1} \in \mathcal{A}(\mathbf{x}_{t'})$,

$$|\mathcal{Q}_{t'}(\mathbf{x}_{t'+1}, d_{t'}) - Q_{t'}^*(\mathbf{x}_{t'+1}, d_{t'})| \leq \lambda(H-t') \ . \tag{B.20}$$

Proof. We will give a proof by induction on t that $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \leq \lambda(H-t)$ for all tuples $\langle t, \mathbf{x}_{t+1}, d_t \rangle$ generated at stage $t = t' \ldots, H-1$ by (4.6) to compute $\mathcal{V}_{t'}(d_{t'})$.

When $t = H-1$, $\mathcal{U}_t(\mathbf{x}_{t+1}, d_t) = \mathcal{Q}_t(\mathbf{x}_{t+1}, d_t)$ in (B.19), by definition. So, (B.20) holds for the base case. Supposing (B.20) holds for $t+1$ (i.e. induction hypothesis), we will prove that it holds for $t = t', \ldots, H-2$:

$$\begin{aligned}
&|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \\
&\leq |\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - \mathcal{U}_t(\mathbf{x}_{t+1}, d_t)| + |\mathcal{U}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \\
&\leq |\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - \mathcal{U}_t(\mathbf{x}_{t+1}, d_t)| + \lambda \\
&\leq \lambda(H-t-1) + \lambda \\
&= \lambda(H-t)
\end{aligned}$$

where the first and the second inequalities follow, respectively, from the triangle inequality and (B.19), and the last inequality is due to

$$\begin{aligned}
&|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - \mathcal{U}_t(\mathbf{x}_{t+1}, d_t)| \\
&\leq \frac{1}{N} \sum_{\ell=1}^{N} |\mathcal{V}_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) - V_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle)| \\
&\leq \frac{1}{N} \sum_{\ell=1}^{N} \max_{\mathbf{x}_{t+2} \in \mathcal{A}(\mathbf{x}_{t+1})} |\mathcal{Q}_{t+1}(\mathbf{x}_{t+2}, \langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) - Q_{t+1}^*(\mathbf{x}_{t+2}, \langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle)| \\
&\leq \lambda(H-t-1)
\end{aligned} \tag{B.21}$$

where the first inequality is due to triangle inequality and the last inequality follows from induction hypothesis.

Finally, when $t = t'$, $|\mathcal{Q}_{t'}(\mathbf{x}_{t'+1}, d_{t'}) - Q_{t'}^*(\mathbf{x}_{t'+1}, d_{t'})| \leq \lambda(H-t')$ (B.20) for all $\mathbf{x}_{t'+1} \in \mathcal{A}(\mathbf{x}_{t'})$ since $d_t = d_{t'}$. □

Proof of Theorem 4.1. It follows immediately from Lemmas B.3 and B.4 that the probability of $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \leq \lambda H$ for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ is at least

$1 - 2(NA)^H \exp(-N\lambda^2/(2K^2))$ where K is previously defined in Lemma B.2.

To guarantee that the probability of $|\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \leq \lambda H$ for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ is at least $1 - \delta$, the value of N has to satisfy the following inequality:

$$1 - 2\left(NA|\right)^H \exp\left(-\frac{N\lambda^2}{2K^2}\right) \geq 1 - \delta \ ,$$

which is equivalent to

$$N \geq \frac{2K^2}{\lambda^2}\left(H \log N + H \log\left(A\right) + \log\frac{2}{\delta}\right). \tag{B.22}$$

Using the identity $\log N \leq \nu N - \log \nu - 1$ for $\nu = \lambda^2/(4K^2H)$, the RHS of (B.22) can be bounded from above by

$$\frac{N}{2} + \frac{2K^2}{\lambda^2}\left(H \log\left(\frac{4K^2HA}{e\lambda^2}\right) + \log\frac{2}{\delta}\right).$$

Therefore, to satisfy (B.22), it suffices to determine the value of N such that

$$N \geq \frac{N}{2} + \frac{2K^2}{\lambda^2}\left(H \log\left(\frac{4K^2HA}{e\lambda^2}\right) + \log\frac{2}{\delta}\right)$$

by setting

$$N = \frac{4K^2}{\lambda^2}\left(H \log\left(\frac{4K^2HA}{e\lambda^2}\right) + \log\frac{2}{\delta}\right)$$

where K is previously defined in Lemma B.2. By assuming H, σ_y^2, and σ_n^2 as constants,

$$N = \mathcal{O}\left(\frac{\kappa^{2H}}{\lambda^2}\log\left(\frac{\kappa A}{\delta\lambda}\right)\right).$$

B.1.5 Approximation Quality of $\mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)$ (4.8)

Proof of Theorem 4.2 Similar to (B.8), the following intermediate function is introduced:

$$\mathbb{U}_t(\mathbf{x}_{t+1}, d_t) \triangleq R(\mathbf{x}_{t+1}, d_t) + V_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mu_t(\mathbf{x}_{t+1})\rangle). \tag{B.23}$$

for $t = 0, \ldots, H-1$.

We will first bound $|Q_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{U}_t(\mathbf{x}_{t+1}, d_t)|$:

$$
\begin{aligned}
&|Q_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{U}_t(\mathbf{x}_{t+1}, d_t)| \\
&= \left| \int_{\mathbb{R}^\kappa} \left(V_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle) - V_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mu_t(\mathbf{x}_{t+1}) \rangle) \right) p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}, d_t) \, \mathrm{d}\mathbf{y}_{t+1} \right| \\
&\leq L_{t+1}(\mathbf{x}_{1:t+1}) \int_{\mathbb{R}^\kappa} \| \mathbf{y}_{t+1} - \mu_t(\mathbf{x}_{t+1}) \| \, p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}, d_t) \, \mathrm{d}\mathbf{y}_{t+1} \\
&= L_{t+1}(\mathbf{x}_{1:t+1}) \int_{\mathbb{R}^\kappa} \| \Psi \mathbf{x}_{t+1} \| \frac{1}{|\Psi|} p(\mathbf{z}_{t+1}) \left| \frac{\partial \mathbf{y}_{t+1}}{\partial \mathbf{z}_{t+1}} \right| \mathrm{d}\mathbf{x}_{t+1} \\
&= L_{t+1}(\mathbf{x}_{1:t+1}) \int_{\mathbb{R}^\kappa} \| \Psi \mathbf{x}_{t+1} \| \, p(\mathbf{z}_{t+1}) \, \mathrm{d}\mathbf{x}_{t+1} \\
&\leq L_{t+1}(\mathbf{x}_{1:t+1}) \, \| \Psi \|_F \, \mathbb{E}_{\mathbf{z}_{t+1}}[\| \mathbf{z}_{t+1} \|] \\
&= L_{t+1}(\mathbf{x}_{1:t+1}) \sqrt{\mathrm{Tr}(\Sigma_{\mathbf{x}_{t+1} | \mathbf{x}_{1:t}})} \, \mathbb{E}_{\mathbf{z}_{t+1}}[\| \mathbf{z}_{t+1} \|] \\
&= \mathcal{O}(\kappa^{H-t} \sqrt{H!/(t+1)!} \, \sigma_n (1 + \sigma_y^2/\sigma_n^2)^{H-t-1/2}) \, \mathbb{E}_{\mathbf{z}_{t+1}}[\| \mathbf{z}_{t+1} \|] \\
&= \mathcal{O}(\kappa^{H-t+1/2} \sqrt{H!/(t+1)!} \, \sigma_n (1 + \sigma_y^2/\sigma_n^2)^{H-t-1/2})
\end{aligned}
\tag{B.24}
$$

where the first equality is due to (4.5) and (B.23), the first inequality is a direct consequence of Theorem B.1 in Appendix B.1.3, the second equality follows from (B.12), $p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}, d_t) = p(\mathbf{z}_{t+1} = \Psi^{-1}(\mathbf{y}_{t+1} - \mu_t(\mathbf{x}_{t+1})))/|\Psi|$ (see Section 35.1.2 in [Taboga, 2017]), and an integration by substitution for multiple variables, the third equality is due to $|\partial \mathbf{y}_{t+1} / \partial \mathbf{z}_{t+1}| = |\Psi|$, the second inequality is due to the submultiplicativity of the Frobenius norm (see Section II.2.1 in [Stewart and Sun, 1990]), the fourth equality follows from a property of Frobenius norm (see Section 10.4.3 in [Petersen and Pedersen, 2012]), the second last equality is due to (B.16), and the last equality follows from $\mathbb{E}_{\mathbf{z}_{t+1}}[\| \mathbf{z}_{t+1} \|] \leq \sqrt{\kappa}$ (see Section 3.1 in [Chandrasekaran *et al.*, 2012]).

We will now give a proof by induction on t that

$$
|Q_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \leq \theta_t \tag{B.25}
$$

for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ where

$$
\theta_t \triangleq \mathcal{O}(\kappa^{H-t+1/2} \sqrt{H!/(t+1)!} \, \sigma_n (1 + \sigma_y^2/\sigma_n^2)^{H-t-1/2}) \, . \tag{B.26}
$$

When $t = H - 1$, $Q_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t) = 0$. So, (B.25) holds for the base case. Supposing (B.25) holds for $t + 1$ (i.e. induction hypothesis), we will prove that it

holds for $t = 0, \ldots, H-2$:

$$
\begin{aligned}
&|Q_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \\
&\leq |Q_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{U}_t(\mathbf{x}_{t+1}, d_t)| + |\mathbb{U}_t(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \\
&\leq \mathcal{O}(\kappa^{H-t+1/2}\sqrt{H!/(t+1)!}\ \sigma_n(1+\sigma_y^2/\sigma_n^2)^{H-t-1/2}) \\
&\quad + |V_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mu_t(\mathbf{x}_{t+1})\rangle) - \mathbb{V}_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mu_t(\mathbf{x}_{t+1})\rangle)| \\
&\leq \mathcal{O}(\kappa^{H-t+1/2}\sqrt{H!/(t+1)!}\ \sigma_n(1+\sigma_y^2/\sigma_n^2)^{H-t-1/2}) \\
&\quad + \max_{\mathbf{x}_{t+2} \in \mathcal{A}(\mathbf{x}_{t+1})} |Q_{t+1}^*(\mathbf{x}_{t+2}, \langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mu_t(\mathbf{x}_{t+1})\rangle) - \mathbb{Q}_{t+1}(\mathbf{x}_{t+2}, \langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mu_t(\mathbf{x}_{t+1})\rangle)| \\
&\leq \mathcal{O}(\kappa^{H-t+1/2}\sqrt{H!/(t+1)!}\ \sigma_n(1+\sigma_y^2/\sigma_n^2)^{H-t-1/2}) + \theta_{t+1} \\
&= \mathcal{O}(\kappa^{H-t+1/2}\sqrt{H!/(t+1)!}\ \sigma_n(1+\sigma_y^2/\sigma_n^2)^{H-t-1/2}) \\
&= \theta_t
\end{aligned}
\tag{B.27}
$$

where the first inequality is due to triangle inequality, the second inequality is due to (B.24), (4.8), and (B.23), and the last inequality is due to the induction hypothesis.

Finally, by assuming H, σ_y^2, and σ_n^2 as constants, it follows from (B.27) that $\theta \triangleq \max_t \theta_t = \mathcal{O}(\kappa^{H+1/2})$ and Theorem 4.2 follows.

B.1.6 Proof of Theorem 4.3

The following lemmas are needed to prove our main result here:

Lemma B.5. *Suppose that the observations d_t, $H \in \mathbb{Z}^+$, a budget of $\kappa(H-t)$ input locations for $t = 0, \ldots, H-1$, $\delta \in (0,1)$, and $\lambda > 0$ are given. Then, the probability of*

$$|Q_t^*(\pi^*(d_t), d_t) - Q_t^*(\pi^\epsilon(d_t), d_t)| \leq 2\lambda H$$

is at least $1-\delta$ by setting N according to that in Theorem 4.1.

Proof.

$$
\begin{aligned}
&Q_t^*(\pi^*(d_t), d_t) - Q_t^*(\pi^\epsilon(d_t), d_t) \\
&\leq Q_t^*(\pi^*(d_t), d_t) - \mathcal{Q}_t(\pi^\epsilon(d_t), d_t) + \lambda H \\
&\leq \max_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} |\mathcal{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| + \lambda H \\
&\leq \lambda H + \lambda H \\
&= 2\lambda H
\end{aligned}
$$

where the first and last inequalities are due to Theorem 4.1 and the second inequality is further due to implication I. □

Lemma B.6. *Suppose that the observations d_t, $H \in \mathbb{Z}^+$, a budget of $\kappa(H-t)$ input*

locations for $t = 0, \ldots, H-1$, $\delta \in (0,1)$, *and* $\lambda > 0$ *are given. Then,*

$$Q_t^*(\pi^*(d_t), d_t) - \mathbb{E}_{\pi^\epsilon(d_t)}[Q_t^*(\pi^\epsilon(d_t), d_t)] \leq 2\lambda H + 4\delta\theta$$

where θ *is previously defined in Theorem 4.2.*

Proof. By Lemma B.5, the probability of $|Q_t^*(\pi^*(d_t), d_t) - Q_t^*(\pi^\epsilon(d_t), d_t)| \leq 2\lambda H$ is at least $1-\delta$. Otherwise, the probability of $|Q_t^*(\pi^*(d_t), d_t) - Q_t^*(\pi^\epsilon(d_t), d_t)| > 2\lambda H$ is at most δ. In the latter case,

$$\begin{aligned}
&|Q_t^*(\pi^*(d_t), d_t) - Q_t^*(\pi^\epsilon(d_t), d_t)| \\
&\leq |Q_t^*(\pi^*(d_t), d_t) - Q_t^\epsilon(\pi^\epsilon(d_t), d_t)| + |Q_t^\epsilon(\pi^\epsilon(d_t), d_t) - Q_t^*(\pi^\epsilon(d_t), d_t)| \\
&\leq \max_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} |Q_t^\epsilon(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| + \lambda H + 2\theta \\
&\leq \lambda H + 2\theta + \lambda H + 2\theta \\
&= 2\lambda H + 4\theta
\end{aligned} \tag{B.28}$$

where the first inequality is due to triangle inequality and the last two inequalities are due to implication II. Recall that π^ϵ is a stochastic policy due to its use of stochastic sampling in $\mathcal{Q}_t$ (4.6), which implies that $\pi^\epsilon(d_t)$ is a random variable. Then,

$$\begin{aligned}
&Q_t^*(\pi^*(d_t), d_t) - \mathbb{E}_{\pi^\epsilon(d_t)}[Q_t^*(\pi^\epsilon(d_t), d_t)] \\
&= \mathbb{E}_{\pi^\epsilon(d_t)}[Q_t^*(\pi^*(d_t), d_t) - Q_t^*(\pi^\epsilon(d_t), d_t)] \\
&\leq (1-\delta)(2\lambda H) + \delta(2\lambda H + 4\theta) \\
&= 2\lambda H + 4\delta\theta
\end{aligned}$$

where the expectation is with respect to random variable $\pi^\epsilon(d_t)$ and the inequality follows from Lemma B.5 and (B.28). □

Proof of Theorem 4.3. We will give a proof by induction on t that

$$V_t^*(d_t) - \mathbb{E}_{\pi^\epsilon}[V_t^{\pi^\epsilon}(d_t)] \leq (2\lambda H + 4\delta\theta)(H - t) \ . \tag{B.29}$$

When $t = H-1$ (i.e., base case),

$$\begin{aligned}
&V_{H-1}^*(d_{H-1}) - \mathbb{E}_{\pi^\epsilon}[V_{H-1}^{\pi^\epsilon}(d_{H-1})] \\
&= Q_{H-1}^*(\pi^*(d_{H-1}), d_{H-1}) - \mathbb{E}_{\pi^\epsilon}[Q_t^{\pi^\epsilon}(\pi^\epsilon(d_{H-1}), d_{H-1})] \\
&= Q_{H-1}^*(\pi^*(d_{H-1}), d_{H-1}) - \mathbb{E}_{\pi^\epsilon(d_{H-1})}[R(\pi^\epsilon(d_{H-1}), d_{H-1})] \\
&= Q_{H-1}^*(\pi^*(d_{H-1}), d_{H-1}) - \mathbb{E}_{\pi^\epsilon(d_{H-1})}[Q_t^*(\pi^\epsilon(d_{H-1}), d_{H-1})] \\
&\leq 2\lambda H + 4\delta\theta
\end{aligned}$$

where the first equality is due to (4.3) and (4.5), the second equality is due to (4.3), the third equality is due to (4.5), and the inequality is due to Lemma B.6. So, (B.29)

holds for the base case. Supposing (B.29) holds for $t+1$ (i.e., induction hypothesis), we will prove that it holds for $t = 0, \ldots, H-2$:

$$
\begin{aligned}
&V_t^*(d_t) - \mathbb{E}_{\pi^\epsilon}[V_t^{\pi^\epsilon}(d_t)] \\
&= Q_t^*(\pi^*(d_t), d_t) - \mathbb{E}_{\pi^\epsilon}[Q_t^{\pi^\epsilon}(\pi^\epsilon(d_t), d_t)] \\
&= Q_t^*(\pi^*(d_t), d_t) - \mathbb{E}_{\pi^\epsilon}[Q_t^*(\pi^\epsilon(d_t), d_t)] + \mathbb{E}_{\pi^\epsilon}[Q_t^*(\pi^\epsilon(d_t), d_t)] - \mathbb{E}_{\pi^\epsilon}[Q_t^{\pi^\epsilon}(\pi^\epsilon(d_t), d_t)] \\
&= Q_t^*(\pi^*(d_t), d_t) - \mathbb{E}_{\pi^\epsilon(d_t)}[Q_t^*(\pi^\epsilon(d_t), d_t)] + \mathbb{E}_{\pi^\epsilon}[Q_t^*(\pi^\epsilon(d_t), d_t) - Q_t^{\pi^\epsilon}(\pi^\epsilon(d_t), d_t)] \\
&\leq 2\lambda H + 4\delta\theta + \mathbb{E}_{\pi^\epsilon}[Q_t^*(\pi^\epsilon(d_t), d_t) - Q_t^{\pi^\epsilon}(\pi^\epsilon(d_t), d_t)] \\
&= 2\lambda H + 4\delta\theta + \mathbb{E}_{\pi^\epsilon}[\mathbb{E}_{\mathbf{y}_{t+1}|\pi^\epsilon(d_t),d_t}[V_{t+1}^*(\langle \mathbf{x}_{1:t} \oplus \pi^\epsilon(d_t), \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle) \\
&\quad - V_{t+1}^{\pi^\epsilon}(\langle \mathbf{x}_{1:t} \oplus \pi^\epsilon(d_t), \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle)]] \\
&= 2\lambda H + 4\delta\theta + \mathbb{E}_{\pi^\epsilon(d_t)}[\mathbb{E}_{\mathbf{y}_{t+1}|\pi^\epsilon(d_t),d_t}[V_{t+1}^*(\langle \mathbf{x}_{1:t} \oplus \pi^\epsilon(d_t), \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle) \\
&\quad - \mathbb{E}_{\pi^\epsilon}[V_{t+1}^{\pi^\epsilon}(\langle \mathbf{x}_{1:t} \oplus \pi^\epsilon(d_t), \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle)]]] \\
&\leq 2\lambda H + 4\delta\theta + \mathbb{E}_{\pi^\epsilon(d_t)}[\mathbb{E}_{\mathbf{y}_{t+1}|\pi^\epsilon(d_t),d_t}[(2\lambda H + 4\delta\theta)(H-t-1)]] \\
&= (2\lambda H + 4\delta\theta)(H-t)
\end{aligned}
\tag{B.30}
$$

where the first and fourth equalities are due to (4.3) and (4.5), the first inequality is due to Lemma B.6, and the last inequality is due to the induction hypothesis.

From (B.30), when $t = 0$,

$$V_0^*(d_0) - \mathbb{E}_{\pi^\epsilon}[V_0^{\pi^\epsilon}(d_0)] \leq 2H(\lambda H + 2\delta\theta) \ .$$

Let $\epsilon = 2H(\lambda H + 2\delta\theta)$ by setting $\lambda = \epsilon/(4H^2)$ and $\delta = \epsilon/(8\theta H)$. Consequently, using Lemma B.5 and $\theta = \mathcal{O}(\kappa^{H+1/2})$ previously defined in Theorem 4.2,

$$N = \mathcal{O}\left(\frac{\kappa^{2H}}{\epsilon^2} \log \frac{\kappa A}{\epsilon}\right) \ .$$

B.1.7 Theoretical analysis of anytime ϵ-Macro-BO

Our result below proves that $\overline{V}_t^*(d_t)$ and $\underline{V}_t^*(d_t)$, which are previously defined in lines 15-16 and 34-35 in Algorithm 4, are upper and lower heuristic bounds of $V_t^*(d_t)$, respectively:

Theorem B.2. *Suppose that the observations $d_{t'}$, $H \in \mathbb{Z}^+$, a budget of $\kappa(H-t')$ input locations for $t' = 0, \ldots, H-1$, $\delta \in (0,1)$, and $\lambda > 0$ are given. Then, the probability of*

$$\underline{V}_t^*(d_t) \leq V_t^*(d_t) \leq \overline{V}_t^*(d_t) \tag{B.31}$$

for all tuples $\langle t, d_t \rangle$ generated at stage $t = t', \ldots, H$ by Algorithm 4 is at least $1-\delta$ by setting N according to Theorem 4.1.

Proof. We will give a proof by induction on t that the probability of (B.31) for all tuples $\langle t, d_t \rangle$ generated at stage $t = t', \ldots, H$ by Algorithm 4 is at least $1-\delta$. The base

case of $t = H$ is true since $\underline{V}^*_H(d_H) = V^*_H(d_H) = \overline{V}^*_H(d_H) = 0$. Supposing (B.31) holds for $t+1$ (i.e. induction hypothesis), we will prove that it holds for $t = t', \ldots, H-1$.

Similar to Lemma B.3 and the main proof of Theorem 4.1, the probability of

$$\mathcal{U}_t(\mathbf{x}_{t+1}, d_t) - \lambda \leq Q^*_t(\mathbf{x}_{t+1}, d_t) \leq \mathcal{U}_t(\mathbf{x}_{t+1}, d_t) + \lambda. \tag{B.32}$$

for all tuples $\langle t, \mathbf{x}_{t+1}, d_t \rangle$ generated at stage $t = t', \ldots, H-1$ by Algorithm 4 is at least $1 - \delta$.

So, the probability of

$$\begin{aligned}
&Q^*_t(\mathbf{x}_{t+1}, d_t) \\
&\leq \mathcal{U}_t(\mathbf{x}_{t+1}, d_t) + \lambda \\
&= R(\mathbf{x}_{t+1}, d_t) + \frac{1}{N} \sum_{\ell=1}^{N} V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) + \lambda \\
&\leq R(\mathbf{x}_{t+1}, d_t) + \frac{1}{N} \sum_{\ell=1}^{N} \overline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) + \lambda \\
&= \overline{Q}^*_t(\mathbf{x}_{t+1}, d_t)
\end{aligned}$$

for all tuples $\langle t, \mathbf{x}_{t+1}, d_t \rangle$ generated at stage $t = t', \ldots, H-1$ by Algorithm 4 is at least $1 - \delta$ where the first inequality follows from (B.32), the first equality is due to definition of $\mathcal{U}_t(\mathbf{x}_{t+1}, d_t)$ (B.8), the last inequality is due to the induction hypothesis, and the last equality is due to definition of $\overline{Q}^*_t$ (see lines 14 and 33 in Algorithm 4). It follows that the probability of $V^*_t(d_t) \leq \overline{V}^*_t(d_t)$ for all tuples $\langle t, d_t \rangle$ generated at stage $t = t', \ldots, H-1$ by Algorithm 4 is at least $1 - \delta$.

Similarly, the probability of

$$\begin{aligned}
&Q^*_t(\mathbf{x}_{t+1}, d_t) \\
&\geq \mathcal{U}_t(\mathbf{x}_{t+1}, d_t) - \lambda \\
&= R(\mathbf{x}_{t+1}, d_t) + \frac{1}{N} \sum_{\ell=1}^{N} V^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) - \lambda \\
&\geq R(\mathbf{x}_{t+1}, d_t) + \frac{1}{N} \sum_{\ell=1}^{N} \underline{V}^*_{t+1}(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) - \lambda \\
&= \underline{Q}^*_t(\mathbf{x}_{t+1}, d_t)
\end{aligned}$$

for all tuples $\langle t, \mathbf{x}_{t+1}, d_t \rangle$ generated at stage $t = t', \ldots, H-1$ by Algorithm 4 is at least $1 - \delta$ where the first inequality is due to (B.32), the first equality is due to definition $\mathcal{U}_t(\mathbf{x}_{t+1}, d_t)$ (B.8), the last inequality is due to the induction hypothesis, and the last equality is due to definition of $\underline{Q}^*_t$ (see lines 13 and 32 in Algorithm 4). It follows that the probability of $V^*_t(d_t) \geq \underline{V}^*_t(d_t)$ for all tuples $\langle t, d_t \rangle$ generated at

stage $t = t', \ldots, H-1$ by Algorithm 4 is at least $1-\delta$. □

Our next result justifies why the function RefineBounds (lines 18-23) in Algorithm 4 can use the tightened heuristic bounds at nodes $\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\bar{\ell}} \rangle$ and $\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^{\ell^*} \rangle$ to refine the heuristic bounds at their siblings (lines 11 and 30) by exploiting the Lipschitz continuity of V^*_{t+1} (Theorem B.1), as explained previously in Section 4.3.1:

Corollary 1. *Suppose that the observations $d_{t'}$, $H \in \mathbb{Z}^+$, a budget of $\kappa(H-t')$ input locations for $t' = 0, \ldots, H-1$, $\delta \in (0,1)$ and $\lambda > 0$ are given. Then, the probability of*

$$\begin{aligned}
&\underline{V}^*_t(\langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1} \oplus \mathbf{y}^j \rangle) - L_t(\mathbf{x}_{1:t}) \|\mathbf{y}^i - \mathbf{y}^j\| \\
&\leq V^*_t(\langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1} \oplus \mathbf{y}^i \rangle) \\
&\leq \overline{V}^*_t(\langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1} \oplus \mathbf{y}^j \rangle) + L_t(\mathbf{x}_{1:t}) \|\mathbf{y}^i - \mathbf{y}^j\|
\end{aligned}$$

between any pair of tuples $\langle t, \langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1} \oplus \mathbf{y}^i \rangle \rangle$ and $\langle t, \langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1} \oplus \mathbf{y}^j \rangle \rangle$ for $i, j = 1, \ldots, N$ generated at stage $t = t'+1, \ldots, H$ by Algorithm 4 is at least $1-\delta$ by setting N according to Theorem 4.1.

Proof.

$$\begin{aligned}
&V^*_t(\langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1} \oplus \mathbf{y}^i \rangle) \\
&\leq V^*_t(\langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1} \oplus \mathbf{y}^j \rangle) + L_t(\mathbf{x}_{1:t}) \|\mathbf{y}^i - \mathbf{y}^j\| \\
&\leq \overline{V}^*_t(\langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1} \oplus \mathbf{y}^j \rangle) + L_t(\mathbf{x}_{1:t}) \|\mathbf{y}^i - \mathbf{y}^j\|
\end{aligned}$$

where the first inequality is a direct consequence of Theorem B.1 in Appendix B.1.3 and the second inequality is due to Theorem B.2.

$$\begin{aligned}
&V^*_t(\langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1} \oplus \mathbf{y}^i \rangle) \\
&\geq V^*_t(\langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1} \oplus \mathbf{y}^j \rangle) - L_t(\mathbf{x}_{1:t}) \|\mathbf{y}^i - \mathbf{y}^j\| \\
&\geq \underline{V}^*_t(\langle \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1} \oplus \mathbf{y}^j \rangle) - L_t(\mathbf{x}_{1:t}) \|\mathbf{y}^i - \mathbf{y}^j\|.
\end{aligned}$$

where the first inequality is a direct consequence of Theorem B.1 in Appendix B.1.3 and the second inequality is due to Theorem B.2. □

Similar to Theorem 4.1, our result below derives a probabilistic guarantee on the approximation quality of $\underline{Q}^*_t(\mathbf{x}_{t+1}, d_t)$:

Theorem B.3. *Suppose that the observations d_t, $H \in \mathbb{Z}^+$, a budget of $\kappa(H-t)$ input locations for $t = 0, \ldots, H-1$, $\delta \in (0,1)$, and $\lambda > 0$ are given and Algorithm 4 terminates at $\omega \triangleq \overline{V}^*_0(d_0) - \underline{V}^*_0(d_0)$ (see line 46 in Algorithm 4). Then, the probability of $|\underline{Q}^*_t(\mathbf{x}_{t+1}, d_t) - Q^*_t(\mathbf{x}_{t+1}, d_t)| \leq 2\lambda + \omega$ for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ is at least $1-\delta$ by setting N according to Theorem 4.1.*

Proof. It follows directly from Theorem B.2 that the probability of

$$|V_0^*(d_0) - \underline{V}_0^*(d_0)| \leq \omega \tag{B.33}$$

is at least $1-\delta$. In general, supposing the planning horizon is reduced to $H-t$ stages for $t = 0, \ldots, H-1$, (B.33) is equivalent to

$$|V_t^*(d_t) - \underline{V}_t^*(d_t)| \leq \omega \tag{B.34}$$

by shifting the indices of $V_0^*(d_0)$ and $\underline{V}_0^*(d_0)$ in (B.33) from 0 to t so that they start at stage t instead. Then, the probability of

$$\begin{aligned}
&|\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \\
&\leq |\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t) - \mathcal{U}_t(\mathbf{x}_{t+1}, d_t)| + |\mathcal{U}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \\
&\leq \lambda + \left| \left(\frac{1}{N} \sum_{\ell=1}^{N} V_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) - \sum_{\ell=1}^{N} \underline{V}_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) \right) + \lambda \right| \\
&\leq 2\lambda + \frac{1}{N} \sum_{\ell=1}^{N} |V_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle) - \underline{V}_{t+1}^*(\langle \mathbf{x}_{1:t+1}, \mathbf{y}_{1:t} \oplus \mathbf{y}^\ell \rangle)| \\
&\leq 2\lambda + \omega
\end{aligned}$$

for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ is at least $1-\delta$ where the first and the third inequalities are due to triangle inequality, the second inequality follows from (B.32), definition of $\mathcal{U}_t(\mathbf{x}_{t+1}, d_t)$ (B.8), and definition of $\underline{Q}_t^*$ (see lines 13 and 32 in Algorithm 4), and the last inequality is due to (B.34). □

We will now formally discuss the implications of the tractable choice of the if condition in (4.12) for theoretically guaranteeing the performance of our $\langle \omega, \epsilon \rangle$-Macro-BO policy $\pi^{\omega\epsilon}$ similarly to that of our ϵ-Macro-BO policy π^ϵ (4.9):

I. In the likely event (with an arbitrarily high probability of at least $1-\delta$) that $|\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \leq 2\lambda + \omega$ for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ (Theorem B.3), $|\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \leq |\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| + |Q_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \leq 2\lambda + \omega + \theta$ for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$, by triangle inequality and Theorems 4.2 and B.3. Consequently, according to (4.12), $Q_t^{\omega\epsilon}(\mathbf{x}_{t+1}, d_t) = \underline{Q}_t^*(\mathbf{x}_{t+1}, d_t)$ for all $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$ and $\pi^{\omega\epsilon}(d_t)$ thus selects the same macro-action as the policy induced by $\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t)$ (see lines 13 and 32 in Algorithm 4).

II. In the unlikely event (with an arbitrarily small probability of at most δ) that $\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t)$ (see lines 13 and 32 in Algorithm 4) is unboundedly far from $Q_t^*(\mathbf{x}_{t+1}, d_t)$ (4.5) (i.e., $|\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| > 2\lambda + \omega$) for some $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$,

$\pi^{\omega\epsilon}(d_t)$ (4.12) guarantees that, for any selected macro-action $\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)$,

$$
\begin{aligned}
&|Q_t^{\omega\epsilon}(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| \\
&= \begin{cases} |\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| & \text{if } |\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \\ & \quad \leq 2\lambda + \omega + \theta, \\ |\mathbb{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| & \text{otherwise;} \end{cases} \\
&\leq \begin{cases} |\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| & \text{if } |\underline{Q}_t^*(\mathbf{x}_{t+1}, d_t) - \mathbb{Q}_t(\mathbf{x}_{t+1}, d_t)| \\ + |\mathbb{Q}_t(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| & \quad \leq 2\lambda + \omega + \theta, \\ \theta & \text{otherwise;} \end{cases} \\
&\leq 2\lambda + \omega + 2\theta \ ,
\end{aligned}
\tag{B.35}
$$

by triangle inequality and Theorem 4.2.

The above implications are central to proving our next result bounding the *expected* performance loss of $\pi^{\omega\epsilon}$ relative to that of Bayes-optimal Macro-BO policy π^*, that is, policy $\pi^{\omega\epsilon}$ is $\langle\omega, \epsilon\rangle$-Bayes-optimal:

Lemma B.7. *Suppose that the observations d_t, $H \in \mathbb{Z}^+$, a budget of $\kappa(H-t)$ input locations for $t = 0, \ldots, H-1$, $\delta \in (0,1)$, and $\lambda > 0$ are given. Then, the probability of*

$$|Q_t^*(\pi^*(d_t), d_t) - Q_t^*(\pi^{\omega\epsilon}(d_t), d_t)| \leq 2\lambda + 2\omega$$

is at least $1-\delta$ by setting N according to that in Theorem 4.1.

Proof.

$$
\begin{aligned}
&Q_t^*(\pi^*(d_t), d_t) - Q_t^*(\pi^{\omega\epsilon}(d_t), d_t) \\
&\leq Q_t^*(\pi^*(d_t), d_t) - \underline{Q}_t^*(\pi^{\omega\epsilon}(d_t), d_t) + 2\lambda + \omega \\
&\leq |Q_t^*(\pi^*(d_t), d_t) - \underline{Q}_t^*(\pi^{\omega\epsilon}(d_t), d_t)| + 2\lambda + \omega \\
&= |Q_t^*(\pi^*(d_t), d_t) - \max_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} \underline{Q}_t^*(\mathbf{x}_{t+1}, d_t)| + 2\lambda + \omega \\
&= |V_t^*(d_t) - \underline{V}_t^*(d_t)| + 2\lambda + \omega \\
&\leq \omega + 2\lambda + \omega \\
&= 2\lambda + 2\omega
\end{aligned}
$$

where the first inequality is due to Theorem B.3, the first equality is further due to implication I discussed just after (4.12), the second equality is due to the definitions of V_t^* (4.5) and $\underline{V}_t^*$ (see lines 15 and 34 in Algorithm 4), and the last inequality is due to (B.34). □

Lemma B.8. *Suppose that the observations d_t, $H \in \mathbb{Z}^+$, a budget of $\kappa(H-t)$ input locations for $t = 0, \ldots, H-1$, $\delta \in (0,1)$, and $\lambda > 0$ are given. Then,*

$$Q_t^*(\pi^*(d_t), d_t) - \mathbb{E}_{\pi^{\omega\epsilon}(d_t)}[Q_t^*(\pi^{\omega\epsilon}(d_t), d_t)] \leq 2\lambda + 2\delta\lambda + 2\omega + 4\delta\theta$$

where θ is previously defined in Theorem 4.2.

Proof. By Lemma B.7, the probability of $|Q_t^*(\pi^*(d_t), d_t) - Q_t^*(\pi^{\omega\epsilon}(d_t), d_t)| \leq 2\lambda + 2\omega$ is at least $1-\delta$. Otherwise, the probability of $|Q_t^*(\pi^*(d_t), d_t) - Q_t^*(\pi^{\omega\epsilon}(d_t), d_t)| > 2\lambda + 2\omega$ is at most δ. In the latter case,

$$\begin{aligned}
&|Q_t^*(\pi^*(d_t), d_t) - Q_t^*(\pi^{\omega\epsilon}(d_t), d_t)| \\
&\leq |Q_t^*(\pi^*(d_t), d_t) - Q_t^{\omega\epsilon}(\pi^{\omega\epsilon}(d_t), d_t)| + |Q_t^{\omega\epsilon}(\pi^{\omega\epsilon}(d_t), d_t) - Q_t^*(\pi^{\omega\epsilon}(d_t), d_t)| \\
&\leq \max_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} |Q_t^{\omega\epsilon}(\mathbf{x}_{t+1}, d_t) - Q_t^*(\mathbf{x}_{t+1}, d_t)| + 2\lambda + \omega + 2\theta \\
&\leq 2\lambda + \omega + 2\theta + 2\lambda + \omega + 2\theta \\
&= 4\lambda + 2\omega + 4\theta
\end{aligned} \tag{B.36}$$

where the first inequality is due to triangle inequality and the last two inequalities are due to (B.35) (i.e., implication II). Recall that $\pi^{\omega\epsilon}$ is a stochastic policy due to its use of stochastic sampling in $\underline{Q}_t^*$(see lines 13 and 32 in Algorithm 4), which implies that $\pi^{\omega\epsilon}(d_t)$ is a random variable. Then,

$$\begin{aligned}
&Q_t^*(\pi^*(d_t), d_t) - \mathbb{E}_{\pi^{\omega\epsilon}(d_t)}[Q_t^*(\pi^{\omega\epsilon}(d_t), d_t)] \\
&= \mathbb{E}_{\pi^{\omega\epsilon}(d_t)}[Q_t^*(\pi^*(d_t), d_t) - Q_t^*(\pi^{\omega\epsilon}(d_t), d_t)] \\
&\leq (1-\delta)(2\lambda + 2\omega) + \delta(4\lambda + 2\omega + 4\theta) \\
&= 2\lambda + 2\delta\lambda + 2\omega + 4\delta\theta
\end{aligned}$$

where the expectation is with respect to random variable $\pi^{\omega\epsilon}(d_t)$ and the inequality follows from Lemma B.7 and (B.36). □

Proof of Theorem 4.4.

We will give a proof by induction on t that

$$V_t^*(d_t) - \mathbb{E}_{\pi^{\omega\epsilon}}[V_t^{\pi^{\omega\epsilon}}(d_t)] \leq (2\lambda + 2\delta\lambda + 2\omega + 4\delta\theta)(H - t) \ . \tag{B.37}$$

When $t = H - 1$ (i.e., base case),

$$\begin{aligned}
&V_{H-1}^*(d_{H-1}) - \mathbb{E}_{\pi^{\omega\epsilon}}[V_{H-1}^{\pi^{\omega\epsilon}}(d_{H-1})] \\
&= Q_{H-1}^*(\pi^*(d_{H-1}), d_{H-1}) - \mathbb{E}_{\pi^{\omega\epsilon}}[Q_t^{\pi^{\omega\epsilon}}(\pi^{\omega\epsilon}(d_{H-1}), d_{H-1})] \\
&= Q_{H-1}^*(\pi^*(d_{H-1}), d_{H-1}) - \mathbb{E}_{\pi^{\omega\epsilon}(d_{H-1})}[R(\pi^{\omega\epsilon}(d_{H-1}), d_{H-1})] \\
&= Q_{H-1}^*(\pi^*(d_{H-1}), d_{H-1}) - \mathbb{E}_{\pi^{\omega\epsilon}(d_{H-1})}[Q_t^*(\pi^{\omega\epsilon}(d_{H-1}), d_{H-1})] \\
&\leq 2\lambda + 2\delta\lambda + 2\omega + 4\delta\theta
\end{aligned}$$

where the first equality is due to (4.3) and (4.5), the second equality is due to (4.3), the third equality is due to (4.5), and the inequality is due to Lemma B.8. So, (B.37) holds for the base case. Supposing (B.37) holds for $t+1$ (i.e., induction hypothesis),

we will prove that it holds for $t = 0, \ldots, H-2$:

$$
\begin{aligned}
&V_t^*(d_t) - \mathbb{E}_{\pi^{\omega\epsilon}}[V_t^{\pi^{\omega\epsilon}}(d_t)] \\
&= Q_t^*(\pi^*(d_t), d_t) - \mathbb{E}_{\pi^{\omega\epsilon}}[Q_t^{\pi^{\omega\epsilon}}(\pi^{\omega\epsilon}(d_t), d_t)] \\
&= Q_t^*(\pi^*(d_t), d_t) - \mathbb{E}_{\pi^{\omega\epsilon}}[Q_t^*(\pi^{\omega\epsilon}(d_t), d_t)] + \mathbb{E}_{\pi^{\omega\epsilon}}[Q_t^*(\pi^{\omega\epsilon}(d_t), d_t)] - \mathbb{E}_{\pi^{\omega\epsilon}}[Q_t^{\pi^{\omega\epsilon}}(\pi^{\omega\epsilon}(d_t), d_t)] \\
&= Q_t^*(\pi^*(d_t), d_t) - \mathbb{E}_{\pi^{\omega\epsilon}(d_t)}[Q_t^*(\pi^{\omega\epsilon}(d_t), d_t)] + \mathbb{E}_{\pi^{\omega\epsilon}}[Q_t^*(\pi^{\omega\epsilon}(d_t), d_t) - Q_t^{\pi^{\omega\epsilon}}(\pi^{\omega\epsilon}(d_t), d_t)] \\
&\leq 2\lambda + 2\delta\lambda + 2\omega + 4\delta\theta + \mathbb{E}_{\pi^{\omega\epsilon}}[Q_t^*(\pi^{\omega\epsilon}(d_t), d_t) - Q_t^{\pi^{\omega\epsilon}}(\pi^{\omega\epsilon}(d_t), d_t)] \\
&= 2\lambda + 2\delta\lambda + 2\omega + 4\delta\theta \\
&\quad + \mathbb{E}_{\pi^{\omega\epsilon}}[\mathbb{E}_{\mathbf{y}_{t+1}|\pi^{\omega\epsilon}(d_t),d_t}[V_{t+1}^*(\langle \mathbf{x}_{1:t} \oplus \pi^{\omega\epsilon}(d_t), \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle) - V_{t+1}^{\pi^{\omega\epsilon}}(\langle \mathbf{x}_{1:t} \oplus \pi^{\omega\epsilon}(d_t), \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle)]] \\
&= 2\lambda + 2\delta\lambda + 2\omega + 4\delta\theta \\
&\quad + \mathbb{E}_{\pi^{\omega\epsilon}(d_t)}[\mathbb{E}_{\mathbf{y}_{t+1}|\pi^{\omega\epsilon}(d_t),d_t}[V_{t+1}^*(\langle \mathbf{x}_{1:t} \oplus \pi^{\omega\epsilon}(d_t), \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle) - \mathbb{E}_{\pi^{\omega\epsilon}}[V_{t+1}^{\pi^{\omega\epsilon}}(\langle \mathbf{x}_{1:t} \oplus \pi^{\omega\epsilon}(d_t), \mathbf{y}_{1:t} \oplus \mathbf{y}_{t+1} \rangle)] \\
&\leq 2\lambda + 2\delta\lambda + 2\omega + 4\delta\theta + \mathbb{E}_{\pi^{\omega\epsilon}(d_t)}[\mathbb{E}_{\mathbf{y}_{t+1}|\pi^{\omega\epsilon}(d_t),d_t}[(2\lambda + 2\delta\lambda + 2\omega + 4\delta\theta)(H - t - 1)]] \\
&= (2\lambda + 2\delta\lambda + 2\omega + 4\delta\theta)(H - t)
\end{aligned}
\tag{B.38}
$$

where the first and fourth equalities are due to (4.3) and (4.5), the first inequality is due to Lemma B.8, and the last inequality is due to the induction hypothesis.

From (B.38), when $t = 0$,

$$V_0^*(d_0) - \mathbb{E}_{\pi^{\omega\epsilon}}[V_0^{\pi^{\omega\epsilon}}(d_0)] \leq 2H(\lambda + \delta\lambda + \omega + 2\delta\theta) = 2\omega H + 2H(\lambda + \delta\lambda + 2\delta\theta) \ .$$

Let $\epsilon = 2H(\lambda + \delta\lambda + 2\delta\theta)$ by setting $\lambda = 1/(4H/\epsilon + 1/(2\theta))$ and $\delta = \epsilon/(8\theta H)$. Consequently, using Lemma B.7 and $\theta = \mathcal{O}(\kappa^{H+1/2})$ previously defined in Theorem 4.2,

$$N = \mathcal{O}\left(\frac{\kappa^{2H}}{\epsilon^2} \log \frac{\kappa A}{\epsilon}\right).$$

B.1.8 Auxiliary Results

Lemma B.9. *$L_t(\mathbf{x}_{1:t}) = \mathcal{O}(\kappa^{H-t+1/2}\sqrt{H!/t!}(1 + \sigma_y^2/\sigma_n^2)^{H-t})$ for $t = 0, \ldots, H-1$.*

Proof. Using Definition B.1 followed by Lemma B.10,

$$
\begin{aligned}
&L_t(\mathbf{x}_{1:t}) \\
&= \max_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} \sqrt{\kappa}\ \alpha(\mathbf{x}_{1:t+1}) + L_{t+1}(\mathbf{x}_{1:t+1})\sqrt{1 + \alpha(\mathbf{x}_{1:t+1})^2} \\
&= (\sqrt{\kappa} + L_{t+1}(\mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1}^*))\ \mathcal{O}(\kappa\sqrt{t+1}(1 + \sigma_y^2/\sigma_n^2))
\end{aligned}
\tag{B.39}
$$

for $t = 0, \ldots, H-1$ where $\mathbf{x}_{t+1}^* \triangleq \operatorname{argmax}_{\mathbf{x}_{t+1} \in \mathcal{A}(\mathbf{x}_t)} L_{t+1}(\mathbf{x}_{1:t} \oplus \mathbf{x}_{t+1})$.

We will now give a proof by induction on t. When $t = H-1$ (i.e., base case), since $L_H(\mathbf{x}_{1:H}) = 0$ (Definition B.1), it follows from (B.39) that $L_{H-1}(\mathbf{x}_{1:H-1}) = \mathcal{O}(\kappa^{3/2}\sqrt{H}(1+\sigma_y^2/\sigma_n^2))$. Supposing Lemma B.9 holds for $t+1$ (i.e., induction hypoth-

esis), we will prove that it holds for $0 \leq t < H-1$:

$$\begin{aligned}
&L_t(\mathbf{x}_{1:t}) \\
&= (\sqrt{\kappa} + \mathcal{O}(\kappa^{H-t-1/2}\sqrt{H!/(t+1)!}\ (1+\sigma_y^2/\sigma_n^2)^{H-t-1}))\ \mathcal{O}(\kappa\sqrt{t+1}(1+\sigma_y^2/\sigma_n^2)) \\
&= \mathcal{O}(\kappa^{H-t+1/2}\sqrt{H!/t!}\ (1+\sigma_y^2/\sigma_n^2)^{H-t})
\end{aligned}$$

where the first equality follows from (B.39) and the induction hypothesis. □

Lemma B.10. *$\alpha(\mathbf{x}_{1:t+1}) = \mathcal{O}(\kappa\sqrt{t+1}(1+\sigma_y^2/\sigma_n^2))$ for $t = 0, \ldots, H-1$ where the function α is previously defined in Lemma B.1.*

Proof. Let $\Xi\Lambda\Xi^\top$ be an eigendecomposition of the symmetric and positive definite $K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} + \sigma_n^2 I$ where Ξ is a matrix whose columns comprise an orthonormal basis of eigenvectors of $\Sigma_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}}$ and Λ is a diagonal matrix with positive eigenvalues of $K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} + \sigma_n^2 I$. From the definition of the function α in Lemma B.1,

$$\begin{aligned}
&\alpha(\mathbf{x}_{1:t+1})^2 \\
&= \|K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}(K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} + \sigma_n^2 I)^{-1}\|_F^2 \\
&= \|K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}\Xi\Lambda^{-1}\Xi^\top\|_F^2 \\
&= \mathrm{Tr}(K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}\Xi\Lambda^{-1}\Xi^\top\Xi\Lambda^{-1}\Xi^\top K_{\mathbf{x}_{1:t}\mathbf{x}_{t+1}}) \\
&= \mathrm{Tr}(K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}\Xi\Lambda^{-2}\Xi^\top K_{\mathbf{x}_{1:t}\mathbf{x}_{t+1}}) \\
&= \mathrm{Tr}(K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}\Xi(\xi^{-2}I)\Xi^\top K_{\mathbf{x}_{1:t}\mathbf{x}_{t+1}}) - \mathrm{Tr}(K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}\Xi(\xi^{-2}I - \Lambda^{-2})\Xi^\top K_{\mathbf{x}_{1:t}\mathbf{x}_{t+1}}) \\
&\leq \mathrm{Tr}(K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}\Xi(\xi^{-2}I)\Xi^\top K_{\mathbf{x}_{1:t}\mathbf{x}_{t+1}}) \\
&= \xi^{-2}\mathrm{Tr}(K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}K_{\mathbf{x}_{1:t}\mathbf{x}_{t+1}}) \\
&= \xi^{-2}\|K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}\|_F^2 \\
&= \mathcal{O}(\kappa^2(t+1)(1+\sigma_y^2/\sigma_n^2)^2)
\end{aligned} \tag{B.40}$$

where ξ is the smallest eigenvalue in Λ, the second equality is due to $(K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}} + \sigma_n^2 I)^{-1} = \Xi\Lambda^{-1}\Xi^\top$, the third and seventh equalities are due to $\|\Phi\|_F^2 = \mathrm{Tr}(\Phi\Phi^\top)$ for any matrix Φ (see Section 10.4.3 in [Petersen and Pedersen, 2012]), the fourth equality follows from the orthonormality of Ξ, the fifth equality is due to linearity of trace, the inequality is due to the positive semidefinite $(\xi^{-2}I - \Lambda^{-2})$ since ξ^{-2} is the largest eigenvalue in Λ^{-2}, and the last equality follows from (a) $\|K_{\mathbf{x}_{t+1}\mathbf{x}_{1:t}}\|_F^2 = \mathcal{O}(\kappa^2(t+1)(\sigma_y^2+\sigma_n^2)^2)$ since every prior covariance is not more than $\sigma_y^2+\sigma_n^2$ and the length of $\mathbf{x}_{1:t}$ is $\mathcal{O}(\kappa(t+1))$ and (b) $\xi \geq \sigma_n^2$ since $K_{\mathbf{x}_{1:t}\mathbf{x}_{1:t}}$ is positive semidefinite and hence $\xi - \sigma_n^2$ is nonnegative. □

B.2 Additional Experimental Results

B.2.1 Simulated plankton density phenomena

See Table B.1 and Table B.2.

Table B.1: Average normalized[12] output measurements observed by the AUV and simple regrets achieved by the tested BO algorithms after 20 observations.

BO Algorithm	Avg. normalized output measurements	Simple regret
ϵ-M-BO $H = 4$	0.6310 ± 0.0458	1.2500 ± 0.0541
ϵ-M-BO $H = 3$	0.5809 ± 0.0486	1.3303 ± 0.0542
ϵ-M-BO $H = 2$	0.5446 ± 0.0464	1.3651 ± 0.0550
DB-GP-UCB	0.5379 ± 0.0462	1.4612 ± 0.0572
Nonmyopic GP-UCB $H = 4$	0.5719 ± 0.0467	1.3984 ± 0.0537
GP-UCB-PE	0.3635 ± 0.0467	1.4079 ± 0.0568
GP-BUCB	0.3396 ± 0.0486	1.3717 ± 0.0573
q-EI	0.2595 ± 0.0444	1.5104 ± 0.0544
BBO-LP	0.3868 ± 0.0444	1.3666 ± 0.0547

Table B.2: Average normalized[12] output measurements achieved by ϵ-Macro-BO with $H = 2$ and $H = 3$ after 20 observations.

Value of β	$H = 2$	$H = 3$
$\beta = 0.0$	0.5563 ± 0.0446	0.5935 ± 0.0461
$\beta = 0.1$	0.6207 ± 0.0458	0.5842 ± 0.0438
$\beta = 0.3$	0.5357 ± 0.0459	0.5240 ± 0.0446
$\beta = 0.6$	0.4226 ± 0.0471	0.5016 ± 0.0470
$\beta = 1.0$	0.3746 ± 0.0460	0.4052 ± 0.0489
$\beta = 2.0$	0.2843 ± 0.0478	0.3566 ± 0.0491
$\beta = 4.0$	0.1919 ± 0.0498	0.2026 ± 0.0441
$\beta = 10.0$	0.0402 ± 0.0468	0.0569 ± 0.0453

B.2.2 Real-World Traffic Phenomenon

See Tables B.3, B.4 and B.5.

B.2.3 Real-World Temperature Phenomenon

See Table B.6, Table B.7 and Table B.8.

Table B.3: Average normalized[12] output measurements observed by the AV and simple regrets achieved by the tested BO algorithms after 20 observations for the real-world traffic phenomenon (i.e., mobility demand pattern).

BO Algorithm	Avg. normalized output measurements	Simple regret
Anytime ϵ-M-GPO $H = 4$	0.2700 ± 0.1014	1.5423 ± 0.1047
Anytime ϵ-M-GPO $H = 3$	0.2574 ± 0.1019	1.5843 ± 0.0994
Anytime ϵ-M-GPO $H = 2$	0.2357 ± 0.1109	1.7396 ± 0.1179
DB-GP-UCB	0.2108 ± 0.1081	1.7050 ± 0.1212
Nonmyopic GP-UCB $H = 4$	0.2267 ± 0.1134	1.7314 ± 0.1158
GP-UCB-PE	0.0770 ± 0.0808	1.5203 ± 0.1247
GP-BUCB	0.0884 ± 0.0819	1.5177 ± 0.1262
q-EI	0.0007 ± 0.0945	1.7945 ± 0.1515
BBO-LP	-0.0077 ± 0.0957	1.7320 ± 0.1149

Table B.4: Average normalized[12] output measurements achieved by *anytime* ϵ-Macro-GPO with $H = 2, 3$ and varying exploration weights β after 20 observations for the real-world traffic phenomenon (i.e., mobility demand pattern).

Value of β	$H = 2$	$H = 3$
$\beta = 0.0$	0.2357 ± 0.1109	0.2574 ± 0.1019
$\beta = 0.2$	0.2550 ± 0.1032	0.2069 ± 0.0987
$\beta = 0.5$	0.1364 ± 0.0967	0.1174 ± 0.0893
$\beta = 1.0$	0.1429 ± 0.0967	0.0911 ± 0.0772
$\beta = 2.0$	0.1174 ± 0.0843	0.0330 ± 0.0755
$\beta = 4.0$	0.0957 ± 0.0841	0.0403 ± 0.0765
$\beta = 10.0$	0.0944 ± 0.0768	-0.0046 ± 0.0756

Table B.5: Average normalized output measurements observed by the AV and simple regrets achieved by *anytime* ϵ-Macro-GPO with $H = 2, 4$ and 20 randomly selected macro-actions per input region, anytime ϵ-Macro-GPO with $H = 2$ and all available macro-actions (the no. of available macro-actions per input region is enclosed in brackets), and EI with all available macro-actions of length 1 after 20 observations for the real-world traffic phenomenon (i.e., mobility demand pattern).

BO Algorithm	Avg. normalized output measurements	Simple regret
Anytime ϵ-M-GPO $H = 4$ (20)	0.2700 ± 0.1014	1.5423 ± 0.1047
Anytime ϵ-M-GPO $H = 2$ (all)	0.2631 ± 0.0918	1.6427 ± 0.0792
Anytime ϵ-M-GPO $H = 2$ (20)	0.2357 ± 0.1109	1.7396 ± 0.1179
EI (all)	0.1469 ± 0.1084	1.6094 ± 0.0946

Table B.6: Average normalized[12] output measurements observed by the mobile robot and simple regrets achieved by the tested BO algorithms after 20 observations for the real-world temperature phenomenon over the Intel Berkeley Research Lab.

BO Algorithm	Avg. normalized output measurements	Simple regret
Anytime ϵ-M-GPO $H = 4$	0.6371 ± 0.0797	0.4069 ± 0.0723
Anytime ϵ-M-GPO $H = 3$	0.6137 ± 0.0829	0.4285 ± 0.0678
Anytime ϵ-M-GPO $H = 2$	0.5450 ± 0.0951	0.5613 ± 0.0834
DB-GP-UCB	0.4874 ± 0.1017	0.6734 ± 0.0934
Nonm. GP-UCB $H = 4$	0.5708 ± 0.0908	0.5911 ± 0.0886
GP-UCB-PE	0.1377 ± 0.0734	0.6700 ± 0.0758
GP-BUCB	0.2067 ± 0.0758	0.6670 ± 0.0762
q-EI	0.3801 ± 0.1044	0.6868 ± 0.1116
BBO-LP	0.2549 ± 0.0833	0.5168 ± 0.0733

Table B.7: Average normalized[12] output measurements achieved by ϵ-Macro-GPO with $H = 2, 3$ and varying exploration weights β after 20 observations for the real-world temperature phenomenon over the Intel Berkeley Research Lab.

Value of β	$H = 2$	$H = 3$
$\beta = 0.0$	0.5450 ± 0.0951	0.6137 ± 0.0829
$\beta = 1.0$	0.6160 ± 0.0820	0.6047 ± 0.0764
$\beta = 2.0$	0.5565 ± 0.0765	0.5787 ± 0.0786
$\beta = 3.0$	0.3755 ± 0.0670	0.4468 ± 0.0645
$\beta = 4.0$	0.1859 ± 0.0608	0.2294 ± 0.0472

Table B.8: Average normalized output measurements observed by the mobile robot and simple regrets achieved by *anytime* ϵ-Macro-GPO with $H = 2, 4$ and 20 randomly selected macro-actions per input region, anytime ϵ-Macro-GPO with $H = 2$ and all available macro-actions (the no. of available macro-actions per input region is enclosed in brackets), and EI with all available macro-actions of length 1 after 20 observations for the real-world temperature phenomenon over the Intel Berkeley Research Lab.

BO Algorithm	Avg. normalized output measurements	Simple regret
Anytime ϵ-M-GPO $H = 4$ (20)	0.6371 ± 0.0797	0.4069 ± 0.0723
Anytime ϵ-M-GPO $H = 2$ (all)	0.6265 ± 0.0861	0.5119 ± 0.0807
Anytime ϵ-M-GPO $H = 2$ (20)	0.5450 ± 0.0951	0.5613 ± 0.0834
EI (all)	0.4565 ± 0.1051	0.8754 ± 0.0941