On Bayesian Methods for Black-Box Optimization Efficiency, Adaptation and Reliability

Zhang, Yunchuan

Awarding institution:
King's College London

On Bayesian Methods for Black-Box Optimization: Efficiency, Adaptation and Reliability

Yunchuan Zhang

Supervisor: Prof. O. Simeone
Dr. Y. Deng

The Department of Engineering
King's College London

This dissertation is submitted for the degree of
Doctor of Philosophy in Engineering

August 2024

To my great motherland, my parents and grandparents, who always supported me in all means to do what I wanted to do

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Yunchuan Zhang
August 2024

Acknowledgements

As time flies, my student career of over two decades has reached its end in London, a fantastic city where my academic journey starts. The previous six years pursuing two degrees at King's were quite fulfilling, and provided me numerous opportunities to explore both academic communities and wonderful life in Europe, such as attending conferences, participating in technical seminars, and rock climbing in the arctic. This thesis would be impossible without the assistance of so many companions along the journey.

Firstly and foremostly, I would like to express my great gratitude to my primary supervisor, Prof. Osvaldo Simeone, for his continuous guidance and support. I am very grateful that he accepted my intention to pursue PhD studies at our first meeting, and showed confidence in me when spending another year applying for the funding. He taught me not only conceptually but also practically on how to become an independent and active researcher. Osvaldo's dedication and technical supervision on my studies are more than what I could have expected from a supervisor.

Secondly, I would like to thank my colleagues in our King's Communication, Learning and Information Processing (KCLIP) lab. Starting with Dr. Sharu Theresa Jose, who taught me how to carry out serious research projects as well as scientific writing, and also provided me psychological counseling when I was depressed and considering giving up. Then Dr. Yansha Deng who was always supportive on my projects and career plans. Followed with Dr. Ivana Nikoloska, for her constructive suggestions on designing my experimental simulations; Dr. Hong Xing, for her kind help with accommodation in London; Dr. Luca Barbieri, for accompanying me through my hardest period; Dr Nicolas Skatchkovsky and Dr. Rahif Kassab, for their advice on life at King's. And sincere gratitude to other colleagues, Jiechen Chen, Clement Ruah, Dr. Zihang Song, Kfir Cohen, Jiayi Huang, Kai Yu, Meiyi Zhu and Qiushuo Hou, for jointly creating a wonderful atmosphere in our KCLIP lab.

Thirdly, I would like to thank my friends at King's, Yutong Li, Anqi Shi, Wen Shang, Dr. Abin Varghese, Linglong Qian, Xinyang Li, Haozhen Li, Wei Gu, Da Tong, Zhiyun Yan, Nian Wang, Junshuo Liu, Yang Liu and Zehua Tang, for enriching my life in London. And my precious friends in China, Qiyu Hu, Yue Zhang, Mengni Wang, Peng Xu, Xingyuan Gu, Tichao Zhou, Yucheng Tu, Yuliang Wu, Xin Liu, Yuxuan Wang, Siying Chen and Yongxin Liu, for their spiritual support during the hard time of COVID-19.

Last but not least, I wish to express my deepest gratitude to my family, for their dedications and all kinds of support which provide me the opportunities to further explore the world and encourage me to make every aspiration come true.

Abstract

Recent advances in many fields ranging from engineering to natural science, require increasingly complicated optimization tasks in the experiment design, for which the target objectives are generally in the form of black-box functions that are expensive to evaluate. In a common formulation of this problem, a designer is expected to solve the black-box optimization tasks via sequentially attempting candidate solutions and receiving feedback from the system. This thesis considers Bayesian optimization (BO) as the black-box optimization framework, and investigates the enhancements on BO from the aspects of efficiency, adaptation and reliability.

Generally, BO consists of a surrogate model for providing probabilistic inference and an acquisition function which leverages the probabilistic inference for selecting the next candidate solution. Gaussian process (GP) is a prominent non-parametric surrogate model, and the quality of its inference is a critical factor on the optimality performance of BO. In many applications, the inherent randomness of the optimization problem caused by the environment conditions may lead to data distribution shift. To enable efficient and adaptive BO in auxiliary optimization tasks, i.e., reaching sub-optimal performance within limited attempts, transfer Bayesian meta-learning is adopted to generalize the surrogate model to address the data distribution shift.

Subsequently, the efficiency and adaptation of meta-learned BO is investigated on a simulated radio resource management (RRM) problem with discrete search space. While the regularity assumptions in GP hold only for continuous input space, this study also formulates the stochastic multi-armed bandit (MAB) model with Bayesian meta-learning for comparison. To further improve the efficiency and adaptation of both BO and MAB schemes, this study introduces a novel mechanism to configure optimizer models with knowledge transferred from graph-based contextual information built upon dynamic network topology.

Furthermore, this study covers the efficiency improvement of multi-fidelity BO (MFBO) with auxiliary optimization tasks addressed sequentially in a fully online manner. To mitigate the problem of evaluating costly objective functions, multi-fidelity optimization setting assumes the designer has access to approximations of the objective functions rather than directly evaluating true objectives, for which higher fidelity evaluations account for better approximations with larger costs. This work devises a novel information-theoretic acquisition function that balances the need to acquire information about the current task

with the goal of collecting information transferable to future tasks. The knowledge transfer, represented by inter-task latent variables, is implemented via particle-based variational Bayesian updates.

Theoretical studies on reliability of BO in sequential black-box optimization with safety constraints on the search space are also covered in this work. The reliability refers to a formal guarantee that the designer is constrained to limit the number of unsafe solutions that are attempted throughout the optimization process in regardless of the assumptions on the surrogate model and evaluations noise level. Online conformal prediction (CP) is adopted in this study to calibrate the set of safe solutions provided by the surrogate model, and obtain the theoretical guarantee by allowing for an arbitrary, controllable but non-zero, rate of violation of the safety constraint. The proposed method is validated on both synthetic and real-world data.

Contents

List of Figures

List of Tables

List of Symbols

Algebra

$\mathbf{x} \in \mathbb{R}^N$ A real row vector in size N

$\mathbf{x}^{\mathsf{H}}$ Conjugate transpose of vector (or matrix) $\mathbf{x}$

$\mathbf{x}^{\mathsf{T}}$ Transpose of vector (or matrix) $\mathbf{x}$

x A random variable

$|x|$ Absolute value of scalar x

$|\mathbf{x}|$ Determinant of matrix $\mathbf{x}$

$||\mathbf{x}||_1 = \sum_{i=1}^{N} |x_i|$ Manhattan distance, or $\ell 1$ norm of vector $\mathbf{x} \in \mathbb{R}^N$

$||\mathbf{x}||_2 = \sqrt{\sum_{i=1}^{N} |x_i|^2}$ Euclidean norm, or $\ell 2$ norm of vector $\mathbf{x} \in \mathbb{R}^N$

x A scalar

x_i i-th element of real vector $\mathbf{x}$

$x_{i,j}$ (i, j)-th element of real vector $\mathbf{x}$

$\mathrm{tr}(\mathbf{x})$ Trace of matrix $\mathbf{x}$

Functions

$\arg\max_{\mathbf{x}} f(\mathbf{x})$ A vector $\mathbf{x}$ in the set of maximizers of $f(\cdot)$

$\frac{\partial f(\mathbf{x})}{\partial x_i}$ Partial derivative of function $f(\cdot)$ with respect to the i-th element of input $\mathbf{x}$

$\mathbb{1}(\cdot)$ Indicator function

$\mathbb{E}(\cdot)$ Expectation function

$\max_{\mathbf{x}} f(\mathbf{x})$ Optimization problem of maximizing function $f(\cdot)$ over input $\mathbf{x}$

$\nabla_{\mathbf{x}} f(\mathbf{x})$ Gradient of function $f(\cdot)$ with respect to input $\mathbf{x}$

$\|\|f\|\|_{\kappa}$	The RKHS norm of function $f(\cdot)$
$f(\cdot)$	A function
$f(\mathbf{x})$	Output of function $f(\cdot)$ for input $\mathbf{x}$, or function itself
$p(\cdot)$	The probability density function
$\exp(\cdot)$	Exponential function
$\log(\cdot)$	Natural logarithm

General

$(1, ..., N)$	Set of integer numbers from 1 to N
$1 : N$	From 1 to N
$[a, b]$	Closed interval between real numbers a and b
$\approx$	"approximated to"
$\cup$	Union
$\emptyset$	Empty set
$\in$	"element of"
∞	Infinity
$\mathbb{R}$	Real numbers
$\mathbb{R}^{N \times M}$	Set of real number vectors in shape $N \times M$
$\mathbf{I}_N$	Identity matrix of size N
$\mathcal{GP}$	Gaussian process
$\prod_{i=1}^{N} x_i$	Product of all N elements
$\propto$	"directly proportional to"
$\setminus$	Relative complement operator
$\sim$	"distributed as per"
$\subset$	"subset of"
$\sum_{i=1}^{N} x_i$	Sum of all N elements
$\|\mathcal{F}\|$	Cardinality of set $\mathcal{F}$

$\{x_i\}_{i=1}^N$	A set of elements from 1-th to N-th entry, or $\{x_1, ..., x_N\}$
$\mathrm{Pr}(\cdot)$	The probability of a specific objective

Acronyms

3GPP	3rd Generation Partnership Project
AI	Artificial Intelligence
AirComp	over-the-Air Computation
BO	Bayesian Optimization
BPSK	Binary Phase-Shift Keying
BS	Base Station
CDF	Cumulative Distribution Function
CP	Conformal Prediction
CSI	Channel State Information
CUB	Caltech-UCSD Birds
DNN	Deep Neural Network
DP	Differential Privacy
EI	Expected Improvement
ES	Entropy Search
Exp3	Exponential-weight algorithm for Exploration and Exploitation
FL	Federated Learning
FLMC	Federated Langevin Monte Carlo
GP	Gaussian Process
ICM	Intrinsic Coregionalization Model
IMRM	Information Meta-Risk Minimization
IRM	Information Risk Minimization
KDE	Kernel Density Estimator
KG	Knowledge Gradient

KL	Kullback-Leibler
KPI	Key Performance Indicator
LMC	Langevin Monte Carlo
LOS	Line-Of-Sight
MAB	Multi-Armed Bandit
MAP	Maximum A Posteriori
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MES	Max-value Entropy Search
MF-MES	Multi-Fidelity Max-value Entropy Search
MFBO	Multi-Fidelity Bayesian Optimization
MFGP	Multi-Fidelity Gaussian Process
MFT-MES	Multi-Fidelity Transferable Max-value Entropy Search
MIMO	Multiple-Input Multiple-Output
NLOS	Non-Line-Of-Sight
NOMA	Non-Orthogonal Multiple Access
OLPC	Open Loop Power Control
OOD	Out-Of-Distribution
PAC	Probably Approximately Correct
PACOH	PAC-Optimal Hyper-posterior
PFR	Plug Flow Reactor
PI	Probability of Improvement
PUSCH	Physical Uplink Shared Channel
RAN	Radio Access Network
RBF	Radial Basis Function
RKHS	Reproducing Kernel Hilbert Space

RMSE	Root Mean Squared Error
SF	Shadow Fading
SNR	Signal-to-Noise Ratio
SR	Simple Regret
SVGD	Stein Variational Gradient Descent
TS	Thompson Sampling
UCB	Upper Confidence Bound
UE	User Equipment
UMi	Urban Microcellular
VI	Variational Inference
WFEM	Weighted Free Energy Minimization
WMMSE	Weighted Minimum Mean Squared Error

Chapter 1

Introduction

The success of many real-world applications ranging from natural science to engineering critically relies on trials and corresponding system responses. The overall goal is to manipulate a set of variables, namely *candidate solutions*, to achieve a desired objective value of interest. For instance, chemists attempt costly experiments with numerous parametric configurations to improve the yield of industrial process or determine conditions for the preparation of medicinal candidates [154]. In a similar manner, industrial manufacturers search for optimal design parameters, including timing control in medical robots [33], quantum heterodyne detection experimental design [164], and beam management in wireless communication systems [201]. Such problems can collectively be formulated as *black-box* optimizations, for which the objective to be optimized is usually *expensive* to evaluate and the analytical expression of the objective function is unavailable.

One of the promising tools for black-box optimization problems is *Bayesian optimization* (BO) [122] which produces candidate solutions approaching global optimum in limited number of attempts without access to gradient information for the objective functions. In order to adapt BO to complicated systems where the designer is faced with a dynamic environment generating auxiliary optimization tasks, knowledge-transferring paradigm is introduced to generalize the surrogate model that BO uses for candidate solution acquisition process. Moreover, the *efficiency* of BO can be further improved by leveraging information extracted from evaluating cheaper approximations of the target objective function, such that the cost of attempts made to the physical system can be significantly reduced [44].

On the other hand, the *reliability* of BO is reflected on a controllable risk of querying undesirable search spaces imposed by safety critics during the sequential optimization process. Examples include designing effective antibiotics while minimizing the potential risk of severe side effects [171], and optimizing motion controller tracking performance while avoiding system instabilities [86]. Therefore, post-processing mechanism for safety constraint surrogate model is required to provide reliable uncertainty estimates on the inferences that are used to define safe exploration regions.

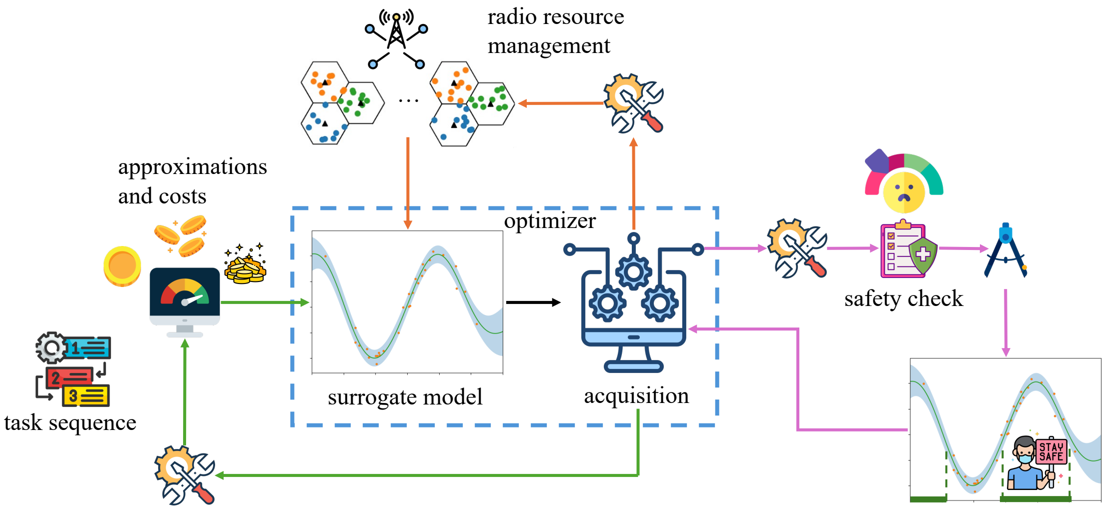

Figure 1.1 Overview of the scope of this work. The common fundamentals of BO algorithms (blue circle) are introduced in Chapter 2 and extensions on applying transfer Bayesian meta-learning are included in Chapter 3. Offline meta-learned BO (orange loop) applied to wireless radio resource management is investigated in Chapter 4. While online sequential multi-task multi-fidelity BO (green loop) based on across-task knowledge transfer is studied in Chapter 5. Finally, safe BO (pink loop) with theoretical guarantee achieved by online conformal prediction is introduced in Chapter 6.

In the following sections, we will first overview the main focus of our study, and then review the literature of BO and meta-learning, which constitutes the basis of this thesis.

1.1 Overview

In this work, we mainly focus on developing Bayesian methods to enable efficient black-box optimization adapted to multiple auxiliary optimization tasks in both online and offline manners, as well as on providing theoretical safety guarantee for online constrained optimization problems.

The corresponding scenarios of interest for our study are outlined as follows:

- As shown in Fig. 1.1, the key component of performing efficient BO on multiple optimization tasks is the surrogate model implementation. While the data distribution for training dataset may vary from target test data distribution, one can seek to improve the capacity of the surrogate model for better generalization. This is the case of interest in Chapter 3, in which we demonstrate how to address data distribution shift via *transfer Bayesian meta-learning*.

- The previous study on generalizing surrogate model provides the basis of efficient BO adapted to multiple optimization tasks. In Chapter 4, we investigate the capability of the *meta-learned* optimizers in a wireless resource allocation problem where the diverse topologies of the mobile devices represent data distribution shift, as seen in orange loop in Fig. 1.1. Furthermore, we will introduce in this work a context-based meta-optimization strategy, in which the mapping from graph-based *contextual information* about the network topology to power allocation parameters is optimized.

- For the case in which the optimization tasks arrive in a *sequence*, as shown in green loop in Fig. 1.1, the across-task knowledge transfer works simultaneously with the candidate solution selection process in an online manner. And in each optimization task, given limited evaluation budget, the optimizer may have access to cheaper approximations to the optimization target, i.e., lower *fidelity* levels. The efficiency of BO in terms of evaluation cost can be further improved by a multi-task multi-fidelity optimization mechanism. In Chapter 5, we will introduce an *information-theoretic* acquisition function that balances the need to acquire information about the current task with the goal of collecting information transferable to future tasks.
- Finally, we turn our focus to the situations where *safety* requirement is considered in the optimization process. With the safety constraints interpreted as black-box functions, the search space for objective function is partitioned into several safe regions, as shown in pink loop in Fig. 1.1. To mitigate the risk of causing harm to the physical system, a *reliable* BO algorithm is required to achieve controllable safety *violation rate* in the optimization process with formal guarantee. This is the case of interest in Chapter 6, we introduce how *online conformal prediction (CP)* calibrates the sets of safe candidate solutions and provides both *deterministic* and *probabilistic* safety guarantee.

1.2 Bayesian Optimization

BO is a popular framework for *expensive-to-evaluate* black-box optimization problems, and it has been widely applied to problems as diverse as biomolecular design [38], chemical experiments design [138], solar irradiance forecasting [118] and automatic detection of bearing localized defect [66]. Notably, BO can provide a more flexible solution that does not require access to gradient information for the objective function and can potentially reduce convergence time as compared to reinforcement learning [112].

Fundamentally, BO consists of two main ingredients: a probabilistic surrogate model that perform Bayesian statistical inference over the unknown objective function to describe the data generation mechanism, and an acquisition function for selecting the next candidate solution to evaluate. After querying the objective function with the chosen candidate, the surrogate model is calibrated to provide a more informative belief over the objective function [152].

The probabilistic surrogate model encodes the current belief of the optimizer about the objective function, and then provides predictions with uncertainty estimates. Gaussian process (GP) [142] is a typical instance of non-parametric probabilistic surrogate model, which places a Gaussian prior distribution over the space of the objective function, and updates to a posterior distribution that depicts the potential output values of querying a candidate input. Alternatively, other common choices of probabilistic surrogate models include scalable models, such as Bayesian neural network (BNN) [87] as well as gradient boosting machines [49]; and likelihood free models based on tree-structured Parzen estimator (TPE) [12].

With the statistical inference provided by the surrogate model, the acquisition function selects the next candidate solution that brings the maximum gain in the considered measure over search space. Specifically, probability of improvement (PI), expected improvement (EI) [74] and knowledge gradient (KG) [46] follow a greedy search policy probing candidate solutions that are likely to improve upon incumbent objective value. In particular, KG along with its variants follow a look-ahead strategy which updates the posterior distribution with hypothetical data and then optimizes the expected gain on the mean estimates. Alternatively, upper confidence bound (UCB) or lower confidence bound (LCB) [161, 162] adopt an optimistic policy that balances between the most uncertain solutions and the current best candidates via calibrated credible intervals. While information-theoretic acquisition functions including entropy search (ES) [60] and Thompson sampling (TS) [177] focus on the posterior distribution over the position of global optimal solution.

1.2.1 Multi-Fidelity BO

BO has been extended to address multi-fidelity – also known as multi-task or multi-information source – settings [125, 172, 141]. The context of multi-fidelity optimization refers to the scenarios where the designer is restricted with some evaluation budget and has access to cheaper approximations of the optimization objective. Via *multi-fidelity BO* (MFBO), information collected at lower fidelity levels can be useful to accelerate the optimization process when viewed as a function of the overall cost budget for evaluating the objective function.

Prior works developed MFBO by building on standard BO acquisition functions, including EI in [91], UCB in [78], and KG in [141, 192]. EI-based MFBO does not account for the level of uncertainty in the surrogate model. This issue is addressed by UCB-based approaches, which, however, require a carefully selected parameter to balance exploitation and exploration. Finally, although KG-based methods achieve efficient global optimization without hyperparameters in the acquisition function, empirical studies in [125] show that they incur a high computational overhead.

In order to mitigate the limitations of the aforementioned standard acquisition functions, reference [172] introduced an information-theoretic acquisition function based on ES. Intuitively, ES-based MFBO seeks for the next candidate solution by maximizing the *information gain per unit cost*. Accordingly, the ES-based acquisition functions aim to reduce the uncertainty about the global optimum, rather than to improve the current best solution as in EI- and KG-based methods, or to explore the most uncertain regions as in UCB-based approaches. However, the key challenge of implementing ES-based methods is the high computation load raised by candidates sampling, as the analytical expressions of the acquisition functions are usually unavailable and approximated by various sampling schemes.

Reference [126] reduced the computation load of ES-based MFBO via *max-value entropy search* (MES) [188]; while the work [174] investigated parallel MFBO extensions. Theoretical and empirical comparisons between a light-weight MF-MES framework and other MFBO approaches are carried out in [125]. As shown recently in [120], the robustness of MF-MES can be guaranteed by introducing a novel mechanism of pseudo-observations when the feedback from lower fidelity levels is unreliable.

In Chapter 5 of this study, we investigate how to perform MFBO with across-task transferable Max-value entropy search for the purpose of tackling multiple successive optimization tasks.

1.2.2 BO with Safety Constraints

Beyond optimizing the objective function, safe exploration in the optimization process is crucial in some applications as well. The context of safety in black-box optimization problems generally refers to imposing some unknown constraint functions over the candidate search space. Accordingly, the constraint functions can also be modelled by independent or joint surrogate models with respect to the objective function.

Existing constrained sequential black-box zero-th order optimizers that leverage BO, collectively referred as *Safe-BO* schemes, target a strict safety requirement whereby no safety violations are allowed. Accordingly, all candidate solutions attempted by the optimizer must be safe [167, 13, 168, 179, 146]. Such stringent safety requirements can only be guaranteed by making strong assumptions on the knowledge available regarding the safety constraint function. In particular, all the existing works on Safe-BO, with a notable exception of [146], either assume knowledge of the smoothness properties of the constraint function when dealing with deterministic constraint function [167, 168, 13, 179], or treating the constraint function as a random realization of a GP with a known kernel when dealing with random constraint function [13].

When the mentioned assumptions or the surrogate model on the constraint function are invalid or infeasible, the existing methods cannot provide any formal safety guarantees. In order to mitigate this problem, reference [146] proposed to apply meta-learning [22] to estimate a suitable surrogate model for the constraint function using additional data that are assumed to be available from other, similar, optimization problems. However, no formal safety guarantees are available for the approach.

In a related line of work, the constrained BO approaches in [54, 62, 53] target a constrained optimization problem, but allow unlimited safety violations during the optimization process for producing an optimal and safe candidate solution at final step. More recent references [114, 199] considered an explicit budget of safety violations for probabilistic constraints, but did not provide any formal safety guarantees.

In Chapter 6 of this study, we introduce a novel BO-based optimization strategy providing *assumptions-free* guarantees on the safety level of the attempted candidate solutions, while enabling any non-zero target safety violation level.

1.3 Meta-Learning

Meta-learning, or learning to learn, is a general paradigm for the design of machine learning algorithms that can transfer shared knowledge from data related to different tasks, to *any* new, related, task. Knowledge is transferred in the form of an optimized inductive bias that can be realized via a prior over the weights of neural networks [5], an initialization of gradient descent [41], or an embedding space shared across auxiliary tasks [181], among other solutions. This paradigm aims to deal with key challenges in many machine learning frameworks, such as restricted availability of training data or computation resources, model generalization, and fast adaptation [65].

Meta-learning is markedly distinct to other knowledge-transferring paradigms such as transfer learning or continual learning. In fact, transfer learning focuses on the optimization of a model for a specific target task given data from a given source task [132]. This yields a pre-trained model with good initializations on the past experience, which can be further fine-tuned on the tasks of interest. In contrast, meta-learning optimizes an adaptation procedure – representing an inductive bias – that can be applied to any, a priori unknown, related task [156].

Knowledge transfer in continual or lifelong learning [184, 133] refers to learning on a stream of tasks generated from a distribution varying over time, with particular focus on fast adaptation in current task as well as without forgetting previous tasks, i.e., catastrophic forgetting. However, the learning objective at meta-level, i.e., generalization and fast adaptation in regardless of non-stationary task distribution, is not explicitly solved in continual learning. To cope with a sequence of tasks, reference [42] introduced an online meta-learning framework based on *follow the leader* mechanism in online learning, and demonstrated a lower theoretical guarantee on the regret performance. Furthermore, the critical concern on the scalability of buffering previous tasks data is solved in [1] via sequentially updating a fixed-size state-vector.

Applications of meta-learning to communication systems are currently limited to deep neural network (DNN) based models, and encompass demodulation [135], channel prediction [136], beamforming [200], feedback design [106], and power control via graph neural networks [131]. We refer to [22] for an extensive review. As shown recently in [146, 130], meta-learning can be combined with BO to achieve convergence and safe exploration within a smaller number of iterations. Applications of this methodology to resource allocation have yet to be explored.

In Chapter 3 and 4 of this study, we first illustrate how to develop transfer Bayesian meta-learning for regression problems, then demonstrate the efficiency and adaptation of meta-learned BO in the wireless radio resource management problem.

1.4 Publication List

This thesis includes the following works, in order of submission time:

[1] **Y. Zhang**, S. T. Jose and O. Simeone, "Transfer Bayesian Meta-Learning Via Weighted Free Energy Minimization," in *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Gold Coast, Australia, 2021.

⇒ Corresponding to Chapter 3

[2] **Y. Zhang**, D. Liu and O. Simeone, "Leveraging Channel Noise for Sampling and Privacy via Quantized Federated Langevin Monte Carlo," in *Proceedings of IEEE International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, Oulu, Finland, 2022.

⇒ Corresponding to Appendix A

[3] **Y. Zhang**, O. Simeone, S. T. Jose, L. Maggi and A. Valcarce, "Bayesian and Multi-Armed Contextual Meta-Optimization for Efficient Wireless Radio Resource Management," in *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 5, pp. 1282-1295, Oct. 2023.

⇒ Corresponding to Chapter 4

[4] **Y. Zhang**, S. Park and O. Simeone, "Bayesian Optimization with Formal Safety Guarantees via Online Conformal Prediction," in *IEEE Journal of Selected Topics in Signal Processing*, pp. 1-15, 2024.

⇒ Corresponding to Chapter 6

[5] **Y. Zhang**, S. Park and O. Simeone, "Multi-Fidelity Bayesian Optimization With Across-Task Transferable Max-Value Entropy Search," *under review at IEEE Transactions on Signal Processing*, 2024.

⇒ Corresponding to Chapter 5

Chapter 2

Preliminaries

In this Chapter, we start with the fundamentals of the probabilistic surrogate model selected in this study – Gaussian process (GP). Afterwards, we brief the standard acquisition functions considered in later chapters.

2.1 Gaussian Process

GP regression is a common statistical approach serving as the probabilistic surrogate model in Bayesian optimization (BO) algorithms. We provide a brief introduction on the basis of GP regression along with a few simple examples in this section. A more comprehensive study on GP can be found in [142].

Consider an unknown scalar-valued function $g(\mathbf{x})$ with input $\mathbf{x} \in \mathbb{R}^d$. GP models such a function by assuming that, for any finite collection $(\mathbf{x}_1,...,\mathbf{x}_N)$ of inputs, the corresponding outputs $(g(\mathbf{x}_1),...,g(\mathbf{x}_1))$ follow a multivariate Gaussian distribution. The Gaussian distribution is characterized by a mean function $\mu(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^d$, and kernel function $\kappa(\mathbf{x},\mathbf{x}')$ for $\mathbf{x},\mathbf{x}' \in \mathbb{R}^d$ [142]. Intuitively, the goal of selected kernel function is to quantify the similarity between any pairs of inputs $\mathbf{x}$ and $\mathbf{x}'$. Larger positive outputs of the kernel function reflect more similar pairs of inputs, encoding the belief that the corresponding function values are closer than those of more diverse input pairs. An example of a kernel function is the radial basis function (RBF) kernel, or Gaussian kernel, represented as

$$\kappa^{(\text{RBF})}(\mathbf{x},\mathbf{x}') = \exp(-h||\mathbf{x}-\mathbf{x}'||_2^2), \tag{2.1}$$

which depends on a lengthscale parameter $h > 0$. Alternatively, one may consider a more general expression of (2.1), namely Matern kernel, expressed as

$$\kappa^{(\text{Matern})}(\mathbf{x},\mathbf{x}') = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(h\sqrt{2\nu}||\mathbf{x}-\mathbf{x}'||_2^2\right)^{\nu} k_{\nu}(h\sqrt{2\nu}||\mathbf{x}-\mathbf{x}'||_2^2), \tag{2.2}$$

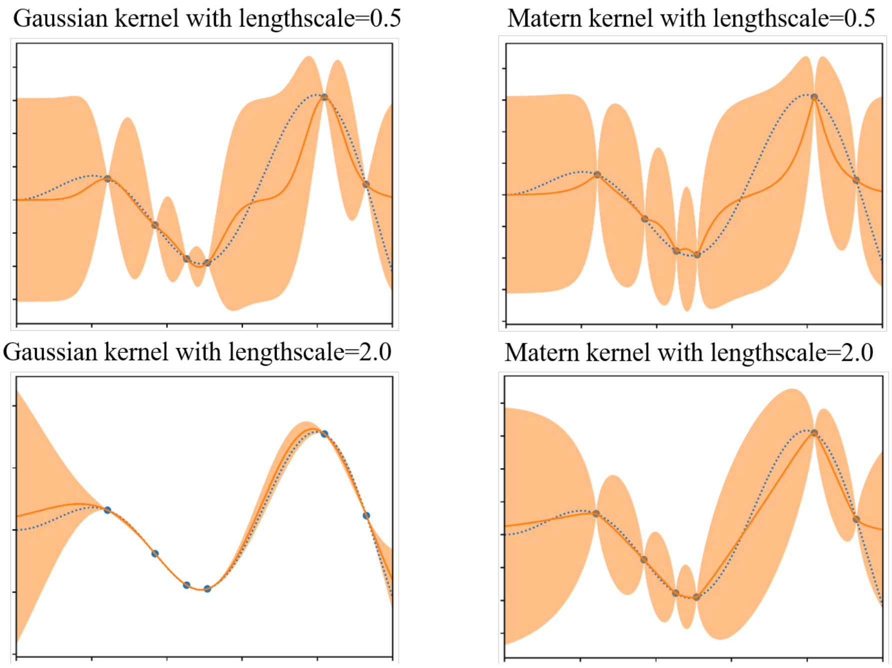

Figure 2.1 GP regression on a single objective function using Gaussian (RBF) kernel and Matern kernel with lengthscale $h = 0.5$ and $h = 2.0$, the smoothness parameter is fixed to be $\nu = 1.5$ for Matern kernel. The blue dashed line is the target objective function, while orange solid lines and shadowing regions represent mean prediction and 95$\%$ confidence intervals provided by GP, respectively.

where parameter ν controls the smoothness of the function; $\Gamma(\cdot)$ is the Gamma function; and $k_\nu(\cdot)$ is a modified Bessel function of the second kind. With the smooth parameter $\nu \rightarrow \infty$, the Matern kernel in (2.2) recovers the RBF kernel in (2.1) with the same lengthscale.

Specifically, for given inputs $(\mathbf{x}_1, ..., \mathbf{x}_N)$, collectively denoted as $\mathbf{X}$, the output vector $(g(\mathbf{x}_1), ..., g(\mathbf{x}_1))$ follows a Gaussian prior distribution $\mathcal{N}(\boldsymbol{\mu}(\mathbf{X}), \mathbf{K}(\mathbf{X}))$, with $N \times 1$ mean vector $\boldsymbol{\mu}(\mathbf{X}) = [\mu(\mathbf{x}_1), ..., \mu(\mathbf{x}_N)]^\mathsf{T}$, and $N \times N$ covariance matrix $\mathbf{K}(\mathbf{X})$ with each (n, n')-th entry given by any selected kernel function $\kappa(\mathbf{x}_n, \mathbf{x}_{n'})$.

Assume that the output $g(\mathbf{x})$ is observed in the presence of independent Gaussian noise as

$$y = g(\mathbf{x}) + \epsilon \tag{2.3}$$

with $\epsilon \sim \mathcal{N}(0, \sigma^2)$, such that, we have the Gaussian likelihood distribution of noisy observation y conditioned on the scalar function output $g(\mathbf{x})$, expressed as

$$p(y|g(\mathbf{x})) = \mathcal{N}(y|g(\mathbf{x}), \sigma^2). \tag{2.4}$$

Observations are modelled as conditionally i.i.d.. Therefore, We write as $\mathbf{y} = [y_1, ..., y_N]^\mathsf{T}$ the $N \times 1$ vector collecting the noisy outputs (2.3) for inputs $(\mathbf{x}_1, ..., \mathbf{x}_N)$, and let $\mathbf{g}(\mathbf{X})$ be the $N \times 1$ vector of outputs of the scalar function $g(\cdot)$, i.e., $\mathbf{g}(\mathbf{X}) = [g(\mathbf{x}_1), \ldots, g(\mathbf{x}_N)]^\mathsf{T}$, we have the conditional distribution

$$p(\mathbf{y}|\mathbf{g}(\mathbf{X})) = \prod_{n=1}^{N} p(y|g(\mathbf{x}_n)). \tag{2.5}$$

An important property of GPs is that, given the history $\mathcal{O} = (\mathbf{X}, \mathbf{y})$ of previous observations $\mathbf{y}$ for inputs $\mathbf{X}$, the posterior distribution $p(g(\mathbf{x})|\mathcal{O})$ of a new output $g(\mathbf{x})$ corresponding to any input $\mathbf{x}$ has a Gaussian distribution with mean $\mu(\mathbf{x}|\mathcal{O})$ and variance $\sigma^2(\mathbf{x}|\mathcal{O})$, i.e.,

$$\begin{aligned} & p(g(\mathbf{x})|\mathcal{O}) = \mathcal{N}(\mu(\mathbf{x}|\mathcal{O}), \sigma^2(\mathbf{x}|\mathcal{O})), && (2.6)\\ \text{with} \quad & \mu(\mathbf{x}|\mathcal{O}) = \mu(\mathbf{x}) + \boldsymbol{\kappa}(\mathbf{x})^{\mathsf{T}}(\mathbf{K}(\mathbf{X}) + \sigma^2 \mathbf{I}_N)^{-1}(\mathbf{y} - \boldsymbol{\mu}(\mathbf{X})), && (2.7)\\ \text{and} \quad & \sigma^2(\mathbf{x}|\mathcal{O}) = \kappa(\mathbf{x}, \mathbf{x}) - \boldsymbol{\kappa}(\mathbf{x})^{\mathsf{T}}(\mathbf{K}(\mathbf{X}) + \sigma^2 \mathbf{I}_N)^{-1}\boldsymbol{\kappa}(\mathbf{x}), && (2.8) \end{aligned}$$

with $N \times 1$ cross-variance vector $\boldsymbol{\kappa}(\mathbf{x}) = [\kappa(\mathbf{x}, \mathbf{x}_1), \ldots, \kappa(\mathbf{x}, \mathbf{x}_N)]^{\mathsf{T}}$ and identity matrix $\mathbf{I}_N \in \mathbb{R}^{N \times N}$. The posterior mean function $\mu(\mathbf{x}|\mathcal{O})$ can be interpreted as a weighted average between prior mean $\mu(\mathbf{x})$ and an estimate of objective values $\mathbf{y}$ with a weight relying on the kernel function. While the posterior variance $\sigma^2(\mathbf{x}|\mathcal{O})$ is represented by the prior covariance after reducing uncertainty due to observing inputs $\mathbf{X}$.

Obviously, the Bayesian statistical inference provided by GP depends on the selection of kernel function as well as the parametric configuration in the kernels. We plot the impacts of kernel type and lengthscale parameter setting on GP regression for a single objective function in Fig. 2.1. With increasing lengthscale parameter h, the function approximated by GP regression is more smooth for both kernel types, while smaller lengthscale values depict more details of the inference. Notably, Matern kernel is preferable for objective functions with abrupt jump, and Gaussian kernel fits well on smooth enough objective functions.

2.2 Bayesian Optimization

Upon the statistical inference provided by GP in (2.6), Bayes optimizer selects the next candidate solution to attempt via optimizing an analytical acquisition function. We briefly introduce three acquisition functions considered as the basis in this section.

2.2.1 Expected Improvement

The greedy search policy based acquisition function – *expected improvement (EI)* is arguably the most common acquisition function in various BO studies. The principle was first introduced in [121], and then applied with GP for optimization in [74]. In the sequential optimization problems, EI computes the average positive increment in the function $g(\mathbf{x}_{n+1})$ evaluated at $\mathbf{x}_{n+1}$ based on the GP inference in (2.6) [71]. Defining as $y_n^* = \max\{y_1, ..., y_n\}$ the current best observed objective value, the EI acquisition function is defined as

$$F(\mathbf{x}|\mathcal{O}) = \Big[\mu(\mathbf{x}|\mathcal{O}) - y_n^* - \xi\Big]\Phi(\delta) + \sigma^2(\mathbf{x}|\mathcal{O})\phi(\delta), \tag{2.9}$$

where

$$\delta = \frac{\mu(\mathbf{x}|\mathcal{O}) - y_n^* - \xi}{\sigma^2(\mathbf{x}|\mathcal{O})}; \tag{2.10}$$

GP mean function $\mu(\mathbf{x}|\mathbf{X}, \tilde{\mathbf{f}})$ and variance function $\sigma^2(\mathbf{x}|\mathbf{X}, \tilde{\mathbf{f}})$ are given as in (2.7) and (2.8), respectively; $\xi \in [0,1)$ is an exploration parameter; and $\Phi(\cdot)$ and $\phi(\cdot)$ are the standard Gaussian cumulative and probability density function, respectively. For a risk-sensitive system with a well-specified GP prior, we may choose small ξ (e.g., $\xi = 0.01$ or even $\xi = 0$). In contrast, where the GP prior is not tailored to the target optimization problem, one can select larger values of ξ to enable more explorations [112].

To this end, BO selects the next candidate solution $\mathbf{x}_{n+1}$ at round $n+1$ via maximizing the EI function in (2.9) over the search space, represented as

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x}|\mathcal{O}). \tag{2.11}$$

In practice, one may adopt gradient descent methods to solve (2.11), e.g., using quasi-Newton method L-BFGS-B as in [103].

However, EI estimates rely on the accuracy of the best observed value so-far, its optimization performance degrades with increasing observation noise power σ^2. In this case, one may consider alternative acquisition functions which quantify the uncertainty in the optimization process.

2.2.2 Upper Confidence Bound

The optimistic policy based acquisition function – *GP upper confidence bound (GP-UCB)* is a common way of negotiating exploration and exploitation in optimization process. It was first proposed in [161], and comprehensively analyzed in [162] with theoretical cumulative regret bound. Let us assume the target optimization problem is a maximization problem (minimization problems correspond to lower confidence bound). GP-UCB computes the upper confidence bound over the function input space based on the GP posterior mean (2.7) and variance (2.8), expressed as

$$\alpha^{(\text{UCB})}(\mathbf{x}|\mathcal{O}) = \mu(\mathbf{x}|\mathcal{O}) + \beta_n\sigma(\mathbf{x}|\mathcal{O}), \tag{2.12}$$

where β_n is a scaling parameter. Similar to the exploration parameter ξ for EI in (2.9), this scaling parameter determines the optimism level of the optimizer on whether exploring most uncertainty input regions or exploiting the high performance candidate solutions. Reference [162] provides theoretical guidelines on updating this scaling parameter to achieve optimal regret performance.

To this end, GP-UCB picks the next candidate solution $\mathbf{x}_{n+1}$ at optimization round $n+1$ by maximizing the optimistic estimate in (2.12) over the entire search space, represented as

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha^{(\mathrm{UCB})}(\mathbf{x}|\mathcal{O}). \tag{2.13}$$

Nevertheless, GP-UCB incorporate strong assumptions on the surrogate model and objective function, e.g., well-specified kernel function, objective function lies in the reproducing kernel Hilbert space (RKHS), and known RKHS norm of the objective. These assumptions are impractical in real-world optimization problems. In Chapter 6, we will detail the potential ways to relax the assumptions in GP-UCB and applications in constrained optimization problems.

2.2.3 Entropy Search

The entropy search (ES) acquisition function, first proposed in [60], follows an information-theoretic method which interacts with the posterior over the location of the unknown global optimum. Specifically, ES computes the mutual information between the global optimum candidate solution $\mathbf{x}^*$ and the objective value $g(\mathbf{x})$ at the next hypothetical candidate solution $\mathbf{x}$. Accordingly, the next attempt is obtained as

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} I(\mathbf{x}^*; g(\mathbf{x})|\mathbf{x},\mathcal{O}) \tag{2.14}$$

$$\text{where} \quad I(\mathbf{x}^*; g(\mathbf{x})|\mathbf{x},\mathcal{O}) = H(\mathbf{x}^*|\mathcal{O}) - \mathbb{E}_{p(g(\mathbf{x})|\mathcal{O})}[H(\mathbf{x}^*|\mathbf{x}, g(\mathbf{x}),\mathcal{O})], \tag{2.15}$$

with $H(\cdot|\cdot)$ representing the differential entropy measure. The key challenge of computing ES acquisition function is the requirement for massive sampling to approximate the second term in (2.15). To mitigate the high computation complexity, reference [61] leverages the symmetry property of mutual information and proposes the predictive entropy search (PES), selecting the next candidate solution via

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} H(g(\mathbf{x})|\mathcal{O}) - \mathbb{E}_{p(\mathbf{x}^*|\mathcal{O})}[H(g(\mathbf{x})|\mathbf{x},\mathbf{x}^*,\mathcal{O})]. \tag{2.16}$$

Unlike ES, this acquisition function computes the differential entropies directly over the GP predictive posterior which can be easily approximated in a closed form.

Furthermore, the max-value entropy search (MES) proposed in [188] introduced a much cheaper and more robust way to implement information-theoretic acquisition functions. Instead of calculating the information gain on the global optimum input view $\mathbf{x}^*$, MES seeks to select candidate solutions that reduce the maximal uncertainty on the global optimal value g^*, thus, the mutual information can be evaluated as

$$I(g^*; g(\mathbf{x})|\mathbf{x}, \mathcal{O}) = H(g(\mathbf{x})|\mathcal{O}) - \mathbb{E}_{p(g^*|\mathcal{O})}[H(g(\mathbf{x})|\mathbf{x}, g^*, \mathcal{O})]. \tag{2.17}$$

The MES acquisition function (2.17) can be easily approximated in an analytical form by treating $p(g(\mathbf{x})|\mathbf{x}, g^*, \mathcal{O})$ as a truncated Gaussian distribution. We will detail the approximation methods and multi-fidelity extensions on MES in Chapter 5.

Chapter 3

Transfer Bayesian Meta-Learning via Weighted Free Energy Minimization

3.1 Overview

In this Chapter, we start by introducing the fundamentals of Bayesian meta-learning and GP that serves as the surrogate model in later chapters on BO. Meta-learning optimizes the hyperparameters of a training procedure, such as its initialization, kernel, or learning rate, based on data sampled from a number of auxiliary tasks. A key underlying assumption is that the auxiliary tasks – known as *meta-training tasks* – share the same generating distribution as the tasks to be encountered at deployment time – known as *meta-test tasks*. This may, however, not be the case when the test environment differ from the meta-training conditions. To address shifts in task generating distribution between meta-training and meta-testing phases, this chapter introduces *weighted free energy minimization* (WFEM) for transfer meta-learning. We instantiate the proposed approach for non-parametric Bayesian regression and classification via GP. The method is validated on a toy example of sinusoidal regression problem, through comparison with standard meta-learning of GP priors as implemented by PACOH [144].

This chapter is organized as follows: in the next section, the context and the problem considered are highlighted. In Sec. 3.3, the fundamentals of GP and the adopted Bayesian meta-learning framework are detailed. We extend the Bayesian meta-learning framework to transfer meta-learning in Sec. 3.4. Experimental results on regression tasks are provided in Sec. 3.5. Finally, Sec. 3.6 concludes the chapter.

3.2 Introduction

3.2.1 Context and Scope

Meta-learning or *learning-to-learn* aims to extract knowledge from a number of auxiliary tasks so as to speed up learning a new, related task [149, 180]. For example, consider the problem of training an image classifier for personalized medical diagnosis on a smart phone. By observing data from other individuals, meta-learning can extract knowledge that allows for a fast adaptation on limited data available for a new user of the service. Information across tasks is shared via meta-learning hyperparameters such as an embedding space shared across tasks [181], an initialization of a neural network [41], or a prior on the weights of a stochastic neural network [5].

An underlying assumption in meta-learning is that the observed auxiliary tasks, known as *meta-training tasks*, and the new, previously unseen *meta-test task* are “related", in the sense that they belong to the same *task environment*. The task environment defines a distribution over the space of data-generating distributions, and the meta-training and meta-test tasks are assumed to be sampled independent identical distributed (i.i.d.) from the same environment [10]. However, this assumption does not hold in many practical scenarios [77]. For instance, in the personalized medical diagnosis example, meta-training data may come from a hospital specializing in patients affected by a specific condition (e.g., cancer patients), while patients at deployment time may not share the same medical history.

A meta-learner trained on tasks from a task environment, such as the hospital in the example above, may not perform well on an out-of-distribution (OOD) meta-test task. Recently, reference [77] introduced the problem of *transfer meta-learning*, which accounts for OOD meta-test tasks by modelling the meta-test environment as being distinct from the meta-training environment. In [77], the authors present PAC Bayes theoretical bounds on the generalization performance of a transfer meta-learner.

In this chapter, a novel transfer Bayesian meta-learning approach is introduced that handles the distribution shift between source task environment and target task environment. The generalization of learning models is obtained by leveraging knowledge extracted from both tasks environments in a transfer learning manner. The proposed method, referred to as WFEM-GP, builds on information meta-risk minimization (IMRM), and is implemented by a non-parametric Bayesian learning model.

3.2.2 Related Work

The problem of meta-learning, when training and testing tasks belong to the same task environment, has been extensively studied both from theoretical perspective [139, 76] and practical applications [137]. The difference between the training and testing environments in meta-learning has been accounted for in the recent works of [25, 94] that develop meta-learning algorithms robust to meta-environment shift. To the best of our knowledge, the work in [77] is the first to formally introduce the problem of transfer meta-learning, and to obtain theoretical generalization performance guarantees for arbitrary meta-learners.

Bayesian approaches to meta-learning have become increasingly popular in the recent years due to their important advantages in quantifying uncertainty and model selection [198]. In this context, both parametric methods such as Bayesian neural networks and non-parametric methods like GPs have both been successfully applied to real-world applications. The works in [144, 45] apply meta-learning to optimize the mean and kernel functions of the GP prior via parametric functions – a method referred to as PACOH-GP. Our work extends PACOH-GP to transfer meta-learning. The recent paper [145] presents a modification of PACOH-GP that operates directly in the functional space.

3.2.3 Main Contribution

Inspired by the theory developed in [77], in this work, we introduce an approach for transfer meta-learning termed WFEM that leverages data from both meta-training and meta-test environments. We specifically focus on non-parametric Bayesian learning via GP, and aim to meta-learn a GP prior to be used on meta-test tasks. WFEM generalizes the IMRM for transfer meta-learning introduced by the theory in [77] to non-parametric base-learners, as well as the PACOH-GP based Bayesian meta-learning approach of [144].

To summarize, the main contributions of this chapter are as follows:

- We introduce the WFEM-GP, which encodes the representations obtained from both meta-training and meta-test environments into non-parametric Bayesian learning models for addressing the OOD problems in meta-learning. The approach is based on constructing a weighted free energy objective to describe the trade-off between different tasks environments.

- As an efficient implementation of WFEM-GP, we propose a particle-based *variational inference* (VI) update strategy for the latent shared parameters by leveraging Stein variational gradient descent (SVGD) [104]

- We demonstrate the advantages of transfer meta-learning over conventional learning and meta-learning schemes. Under *meta-environment shift* between the training and testing task environments, we show that the data from the meta-training environment can help improving the predictive performance of transfer meta-learner on the meta-test task as compared to conventional meta-learning schemes such as PACOH-GP.

3.3 Problem Formulation

In this chapter, we focus on non-parametric Bayesian learning for regression and classification, whereby the prior is meta-learned using data from multiple related tasks. In this section, we review the framework introduced in [144] that defines Bayesian meta-learning as the minimization of a free energy functional. In the next section, we extend this framework to transfer meta-learning by leveraging the theoretical results in [77]. To this end, we first review non-parametric Bayesian learning via GP, and then we describe the problem of meta-learning the GP prior proposed in [144].

3.3.1 Parameterized Gaussian Process

We study supervised learning problems. Accordingly, let $\mathbf{X} = \{\mathbf{x}_m\}_{m=1}^{M}$ denote a set of M inputs in $\mathbb{R}^d$, and let $\mathbf{y} = \{y\}_{m=1}^{M}$ denote the corresponding observation outputs. Each tuple $(\mathbf{x}_m, y_m)$ is assumed to be generated i.i.d. according to an unknown population distribution P. We also denote $\mathcal{D} = \{(\mathbf{x}_m, y_m)\}_{m=1}^{M}$ as the collected training data set, and $\mathbf{f}(\mathbf{X})$ be the $M \times 1$ vector of outputs of the random scalar function $f(\cdot)$, i.e., $\mathbf{f}(\mathbf{X}) = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_M)]^{\mathsf{T}}$.

Unlike the fundamental GP introduced in Sec. 2.1, the GP prior is parameterized in terms of a *hyperparameter vector* $\boldsymbol{\theta}$ that determines the mean function $\mu_{\boldsymbol{\theta}}(\cdot)$ and the kernel function $k_{\boldsymbol{\theta}}(\cdot,\cdot)$. Accordingly, the GP defines a prior joint distribution on the output values $\mathbf{f}(\mathbf{X})$ as

$$p_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{X})) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{X}), \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})), \tag{3.1}$$

where $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{X}) = [\mu_{\boldsymbol{\theta}}(\mathbf{x}_1), ..., \mu_{\boldsymbol{\theta}}(\mathbf{x}_M)]^T$ is the $M \times 1$ mean vector, and $\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})$ represents the $M \times M$ covariance matrix whose (i, j)th entry is given as $[\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})]_{i,j} = k_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j)$.

Using the GP prior introduced in (3.1) and the Gaussian data likelihood distribution $\mathcal{N}(y|f(\mathbf{x}), \sigma^2)$, the posterior distribution of the random function $f(\mathbf{x})$ at a new test input $\mathbf{x}$ can be obtained as [142]

$$\begin{aligned} & p_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathcal{D}) \sim \mathcal{N}(\mu_{\boldsymbol{\theta}}(\mathbf{x}|\mathcal{D}), \sigma^2_{\boldsymbol{\theta}}(\mathbf{x}|\mathcal{D})), && (3.2)\\ \text{where}\quad & \mu_{\boldsymbol{\theta}}(\mathbf{x}|\mathcal{D}) = \mu_{\boldsymbol{\theta}}(\mathbf{x}) + \mathbf{k}_{\mathcal{D}}(\mathbf{x})^{\mathsf{T}}(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}))^{-1}(\mathbf{y} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{X})), && (3.3)\\ & \sigma^2_{\boldsymbol{\theta}}(\mathbf{x}|\mathcal{D}) = k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{\mathcal{D}}(\mathbf{x})^{\mathsf{T}}(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}))^{-1}\mathbf{k}_{\mathcal{D}}(\mathbf{x}), && (3.4) \end{aligned}$$

with the $M \times M$ Gramian matrix $\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}) = \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}) + \sigma^2\mathbf{I}_M$ and $\mathbf{k}_{\mathcal{D}}(\mathbf{x})$ being the $M \times 1$ cross-variance vector $\mathbf{k}_{\mathcal{D}}(\mathbf{x}) = [k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_1), \ldots, k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_M)]^{\mathsf{T}}$.

We will also require the evidence or *marginal likelihood* of the *output labels*,

$$p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{X}) = \int p_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{X}))p(\mathbf{y}|\mathbf{f}(\mathbf{X}))\mathbf{d}\mathbf{f}(\mathbf{X}), \tag{3.5}$$

the log of which can be obtained in closed form for the Gaussian likelihood as [142]

$$\ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}(\mathbf{y}-\mu_{\boldsymbol{\theta}}(\mathbf{X}))^{\mathsf{T}}(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}))^{-1}(\mathbf{y}-\mu_{\boldsymbol{\theta}}(\mathbf{X})) - \frac{1}{2}\ln|\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X})| - \frac{M}{2}\ln 2\pi, \tag{3.6}$$

where $|\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X})|$ denotes the determinant of the Gramian matrix.

3.3.2 Meta-learning the GP Prior (PACOH-GP)

In GP, the hyperparameter vector $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ that describes the mean and kernel functions of the GP in (3.1) is fixed *a priori*, possibly using cross-validation. In contrast, the meta-learning approach introduced in [144] – termed PACOH-GP – aims to automatically infer the hyperparameters $\boldsymbol{\theta}$ of the GP prior (3.1) by observing data from tasks with similar statistical properties [144]. Note that, the kernel function could be parametrized as

$$k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') = \frac{1}{2}\exp\left(-||\Phi_{\boldsymbol{\theta}}(\mathbf{x}) - \Phi_{\boldsymbol{\theta}}(\mathbf{x}')||_2^2\right), \tag{3.7}$$

where $\Phi_{\boldsymbol{\theta}}(\cdot)$ is a parametric function, typically instantiated as a deep neural network (DNN), with $\boldsymbol{\theta}$ constituting its weight and biases.

Following the setting of Baxter [10], the tasks observed by a meta-learner are assumed to be sampled i.i.d from a *task environment*, which defines a distribution P_T over the space of tasks. Precisely, for each i-th observed task τ_i, we sample a data distribution $P_i \sim P_T$ from the task environment, and the corresponding dataset $\mathcal{D}_i = (\mathbf{X}_i, \mathbf{y}_i) = \{\mathbf{x}_{i,m}, \mathbf{y}_{i,m}\}_{m=1}^{M_i}$ of M_i samples are generated i.i.d. according to the unknown population distribution P_i. The dataset generated from observing N meta-training tasks constitutes the *meta-training set* $\mathcal{D}_{1:N} = (\mathcal{D}_1, \ldots, \mathcal{D}_N)$.

At test time, the meta-learner is given data from a *meta-test task* with unknown population distribution P drawn from the task environment P_T. We denote $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$ as the M-sample *meta-test training dataset* generated i.i.d according to P, and $(\mathbf{x}, y)$ as the test data pair sampled from the meta-test task.

Following the non-parametric Bayesian model described in the previous subsection, we model each i-th observed task τ_i through a random scalar function $f_i(\mathbf{x})$. Importantly, all tasks share the same GP prior $\mathcal{GP}(\mu_{\boldsymbol{\theta}}(\mathbf{x}), k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}'))$. Therefore, transfer of the shared information among data points of different tasks takes place through the hyperparameter vector $\boldsymbol{\theta}$.

The goal of meta-learning is to infer the shared hyperparameter vector $\boldsymbol{\theta}$ of the GP prior in (3.1), using the meta-training data $\mathcal{D}_{1:N}$, for use on a new, previously unseen *meta-test* task. The meta-test task is modelled by a random function $f(\mathbf{x}) \sim \mathcal{GP}(\mu_{\boldsymbol{\theta}}(\mathbf{x}), k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}'))$ that share the same hyperparameter vector $\boldsymbol{\theta}$ as the meta-training tasks.

The shared hyperparameter vector $\boldsymbol{\theta}$ is endowed with a *hyper-prior* distribution $p(\boldsymbol{\theta})$, and the meta-learner uses the *meta-training* dataset $\mathcal{D}_{1:N}$ to update the hyper-prior $p(\boldsymbol{\theta})$ to a *hyper-posterior* distribution $q(\boldsymbol{\theta}|\mathcal{D}_{1:N})$ [144]. This is done by minimizing a free energy metric. Specifically, the *meta-training loss* of hyperparameter vector $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ incurred on the meta-training set $\mathcal{D}_{1:N}$ is defined as

$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_{1:N}) = \frac{1}{N} \sum_{i=1}^{N} \frac{-\log p_{\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{X}_i)}{M_i}, \tag{3.8}$$

which is the empirical average of the negative marginal log-likelihood $p_{\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{X}_i)$ across the meta-training tasks. This is defined as in (3.6) for a Gaussian likelihood. The hyper-posterior distribution $q(\boldsymbol{\theta}|\mathcal{D}_{1:N})$ is then optimized so as to minimize the *free energy objective* [75],

$$\mathcal{F}(q) = \mathbb{E}_{q(\boldsymbol{\theta}|\mathcal{D}_{1:N})}[\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_{1:N})] + \gamma^{-1}\mathrm{KL}(q(\boldsymbol{\theta}|\mathcal{D}_{1:N})||p(\boldsymbol{\theta})), \tag{3.9}$$

where $\gamma > 0$ is a *temperature* parameter, and $\mathrm{KL}(p||q)$ denotes the Kullback–Leibler (KL) divergence between the distributions p and q. The choice of the temperature parameter calibrating the hyper-posterior q is investigated in [108, 117, 173].

The meta-free energy function $\mathcal{F}(q)$ is the sum of: (a) the *average meta-training loss* for a randomly drawn hyperparameter vector $\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\mathcal{D}_{1:N})$; and (b) the KL-divergence term between the hyper-posterior $q(\boldsymbol{\theta}|\mathcal{D}_{1:N})$ and the hyper-prior $p(\boldsymbol{\theta})$, which serves as a regularization on the meta-level. Let $\gamma^{-1} = (1/N + 1/\widetilde{M})$, with $\widetilde{M} = (N^{-1}\sum_{n=1}^{N} M_n^{-1})^{-1}$ serve as the harmonic mean, the function $\mathcal{F}(q)$ in (3.9) corresponds to an upper bound (neglecting constant terms independent of distribution q) on the population test log-loss obtained via PAC-Bayes analysis [144]. The optimization problem in (3.9) also corresponds to a form of information risk minimization (IRM) [204] and generalized Bayesian meta-learning [144].

The minimizing solution can be obtained in closed form as the *Gibbs hyper-posterior* distribution [84, 144, 202]

$$q^{\text{PACOH-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N}) \propto p(\boldsymbol{\theta}) \exp\Big(-\gamma \mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_{1:N})\Big). \tag{3.10}$$

During meta-testing, we evaluate the average predictive distribution as

$$\mathbb{E}_{q^{\text{PACOH-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N})}[p_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathcal{D})], \tag{3.11}$$

of the meta-test task function $f(\mathbf{x})$ at its test input $\mathbf{x}$.

3.4 Transfer Meta-learning the GP Prior

In this section, we introduce and formulate the problem of *transfer* meta-learning the GP prior, inspired by the recent theoretical work in [77]. As explained in the previous section, in conventional meta-learning, the meta-training and meta-test tasks belong to the same *task environment* in the sense that the population distributions for each task are drawn from the same task distribution P_T. In contrast, transfer meta-learning concerns settings in which the observed meta-training tasks belong to a *source task environment*, while the meta-test task belongs to a different *target task environment*. The task distributions for the two environments are respectively denoted as P_T^S and P_T^T.

In this work, we assume that the transfer meta-learner, in addition to data from the source task environment, observes data from a limited number of tasks from the target task environment. As such, the meta-training set $\mathcal{D}_{1:N} = (\mathcal{D}_1, \ldots, \mathcal{D}_N)$ comprises of N data sets, of which a subset of βN data sets, for $\beta \in [0,1]$, correspond to tasks from the source task environment, and the remaining from the target task environment. Accordingly, for each task τ_i, $i = 1, \ldots, \beta N$, a data distribution P_i is sampled from the source task environment P_T^S, with the corresponding M_i-sample training data generated i.i.d. according to P_i. For $i = \beta N + 1, \ldots, N$, each task τ_i samples a data distribution P_i from the target task environment P_T^T. The meta-test task is drawn from the target task environment. Let $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ denote the M-sample meta-test training data, and $(\mathbf{x}, y)$ a meta-test test data point.

The goal of transfer meta-learning is to use the observed meta-training dataset $\mathcal{D}_{1:N}$ to infer a hyperparameter vector $\boldsymbol{\theta}$ of the GP prior for use on a new meta-test task. As in [144], the transfer meta-learner uses the meta-training dataset $\mathcal{D}_{1:N}$ to update the hyper-prior distribution $p(\boldsymbol{\theta})$ to a hyper-posterior distribution $q(\boldsymbol{\theta}|\mathcal{D}_{1:N})$.

To this end, the transfer meta-learner considers the following *weighted average meta-training loss*,

$$\bar{\mathcal{L}}(\boldsymbol{\theta}, \mathcal{D}_{1:N}) = \alpha \mathcal{L}_s(\boldsymbol{\theta}, \mathcal{D}_{1:\beta N}) + (1-\alpha) \mathcal{L}_t(\boldsymbol{\theta}, \mathcal{D}_{\beta N+1:N}), \tag{3.12}$$

for $\alpha \in [0,1]$, which is a convex combination of the training loss

$$\mathcal{L}_s(\boldsymbol{\theta}, \mathcal{D}_{1:\beta N}) = \frac{1}{\beta N} \sum_{i=1}^{\beta N} \frac{-\log p_{\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{X}_i)}{M_i} \tag{3.13}$$

evaluated on data from the source environment and of the training loss computed on data from the target environment

$$\mathcal{L}_t(\boldsymbol{\theta}, \mathcal{D}_{\beta N+1:N}) = \frac{1}{(1-\beta)N} \sum_{i=\beta N+1}^{N} \frac{-\log p_{\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{X}_i)}{M_i}. \tag{3.14}$$

In the proposed *weighted free energy minimization with Gaussian Processes (WFEM-GP)*, the target hyper-posterior distribution $q(\boldsymbol{\theta}|\mathcal{D}_{1:N})$ is optimized so as to minimize the *weighted free energy functional*

$$\mathcal{F}(q) = \mathbb{E}_{q(\boldsymbol{\theta}|\mathcal{D}_{1:N})}[\bar{\mathcal{L}}(\boldsymbol{\theta}, \mathcal{D}_{1:N})] + \gamma^{-1}\mathrm{KL}(q(\boldsymbol{\theta}|\mathcal{D}_{1:N})||p(\boldsymbol{\theta})). \tag{3.15}$$

Setting $M_i = M$, for $i = 1, \ldots, N$ and $\gamma^{-1} = (1/N + 1/M)$, the free energy functional in (3.15) (neglecting constant terms) provides a PAC-Bayesian upper bound on the population test log-loss [77]. Similar to (3.10), the minimizing solution is obtained as the *Gibbs hyper-posterior*

$$q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N}) \propto p(\boldsymbol{\theta}) \exp\left[-\gamma\bar{\mathcal{L}}(\boldsymbol{\theta}, \mathcal{D}_{1:N})\right]. \tag{3.16}$$

Finally, the hyper-posterior $q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N})$ is used in lieu of the corresponding PACOH-GP hyper-posterior distribution in (3.11) in order to define the predictive distribution.

In practice, for both PACOH-GP and WFEM-GP, the expectations in the predictive distributions (3.11) need to be approximated. As detailed in the supplementary materials B.1, this can be done by evaluating the maximum of the hyper-posteriors, and by plugging this value into the predictive distribution $p_{\boldsymbol{\theta}}(f(\mathrm{x})|\mathcal{D})$ – an approach we refer to as *maximum a posteriori* (MAP). Alternatively one can use an average obtained via the particle-based inference through *Stein Variational Gradient Descent* (SVGD) [104]. In contrast to *Markov Chain Monte Carlo* (MCMC) schemes with strongly correlated samples and slow convergence, SVGD balances between particle-efficient convergence to the MAP solution and maintaining the diversity of particles to capture the multi-modality of the target posterior $q(\boldsymbol{\theta}|\mathcal{D}_{1:N})$ by introducing a repulsive force term [104].

3.5 Numerical Results

In this section, we detail our experimental settings and compare WFEM-GP and PACOH-GP on a synthetic dataset.

We demonstrate the advantage of WFEM-GP over GP and PACOH-GP by considering a sinusoidal regression problem. Towards this, we first describe the data generation process for source and target environment. For each task, the input x is drawn from a uniform distribution $\mathcal{U}(-5, 5)$. The output y corresponding to an input x is obtained as $y \sim \mathcal{N}(y|f(\mathrm{x}), \sigma^2)$ with standard deviation $\sigma = 0.1$, where

$$f(\mathrm{x}) = \mathrm{a}\mathrm{x} + \mathrm{b}\sin(1.5(\mathrm{x} - \mathrm{c})) + \mathrm{d}, \tag{3.17}$$

and the scalars $\mathrm{a}, \mathrm{b}, \mathrm{c}$ and d characterize a given task. The source (or target) task environment defines a joint distribution over the parameters $(\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d})$. Specifically, each task

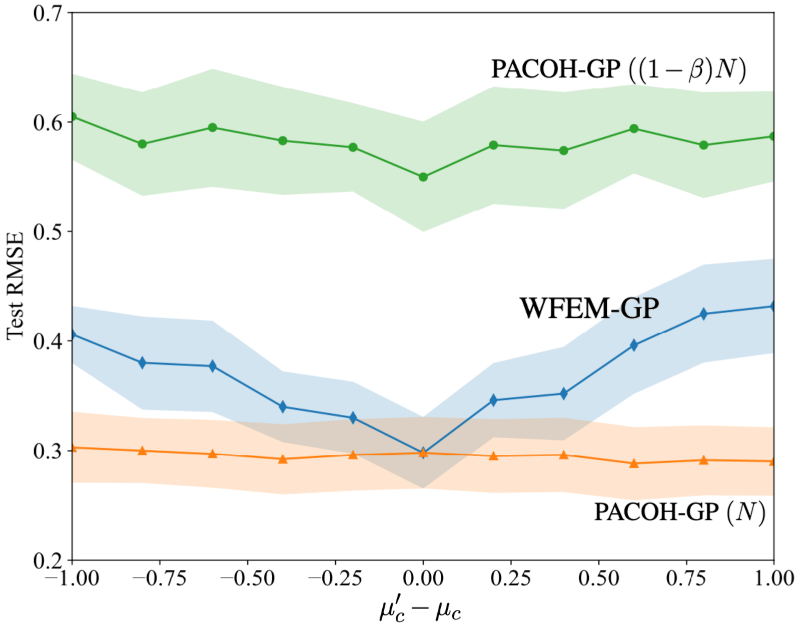

Figure 3.1 Average test root mean square error (RMSE) under three schemes – PACOH-GP with N tasks and with $(1-\beta)N$ tasks from the target environment and WFEM-GP – as a function of the deviation $\mu_c' - \mu_c$ of the target task environment from the source task environment with fixed $\mu_c = 0$.

from the source task environment is sampled as

$$\begin{aligned} &\mathrm{a} \sim \mathcal{N}(0.5, 0.2^2), \mathrm{b} \sim \mathcal{U}(0.7, 1.3), \\ &\mathrm{c} \sim \mathcal{N}(\mu_c, 0.1^2), \mathrm{d} \sim \mathcal{N}(5.0, 0.1^2), \end{aligned} \tag{3.18}$$

where $\mu_c \in \mathbb{R}$ denotes the mean value of the parameter c. We consider the target task environment to follow the same distributions for parameters a, b and d as in (3.18), while the parameter c is distributed as $\mathrm{c} \sim \mathcal{N}(\mu_c', 0.1^2)$, with a mean μ_c' distinct from μ_c in (3.18). We assume a Gaussian likelihood for each i-th task as $p(y|f(\mathbf{x})) = \mathcal{N}(y|f(\mathbf{x}), \sigma^2)$.

We instantiate the mean funtion $\mu_{\boldsymbol{\theta}}(\cdot)$ and kernel function $K_{\boldsymbol{\theta}}(\cdot, \cdot)$ as neural networks, where the hyperparameter $\boldsymbol{\theta}$ corresponds to the weights and biases of the neural networks can be meta-learned. Both neural networks are 4 layered fully-connected neural networks with 32 neurons in each layer and tanh non-linearities. We use adaptive moment estimation (Adam) to optimize the gradient descent of updating hyperparameter $\boldsymbol{\theta}$ for GP. In meta-testing phase, the performance on meta-test datasets is evaluated as detailed in Appendix B.2.

In Fig. 3.1, we compare the performance of WFEM-GP with that of PACOH-GP that uses N tasks or $(1-\beta)N$ tasks from target task environment. We vary the deviation $\mu_c' - \mu_c$ of the mean μ_c' of parameter c in the target environment from a fixed mean μ_c of the source environment. We set $\sigma = 0.1$, $\alpha = \beta = 0.5$, number of tasks $N = 30$, and

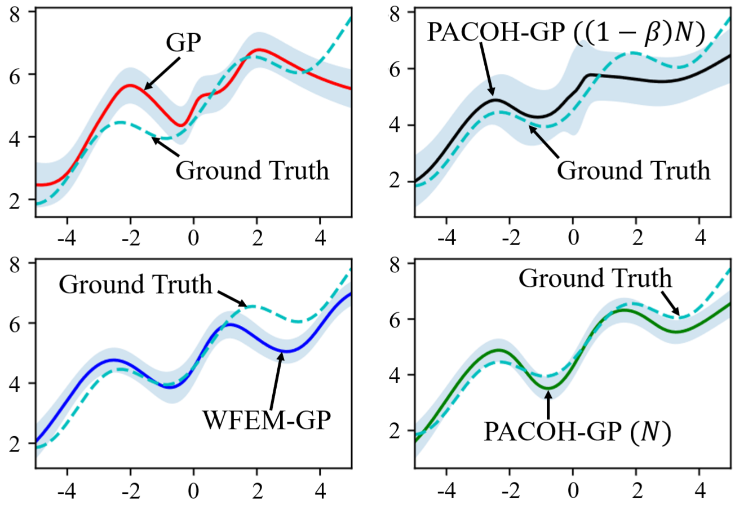

Figure 3.2 Comparison of posterior predictions under the four schemes – GP, PACOH-GP with N tasks and with $(1-\beta)N$ tasks from the target environment and WFEM-GP – against the ground-truth regression function.

number of samples per task $M_i = 5$. We adopt the MAP approximation strategy for all the learning schemes.

WFEM-GP is seen to outperform the PACOH-GP scheme that uses only $(1-\beta)N$ tasks from the target environment. This suggests that data from the source task environment can be utilized during meta-training to improve the performance on test tasks from the target environment. We also benchmark the performance of WFEM-GP against the ideal performance of a meta-learner trained on N target tasks. It can be seen that the transfer meta-learner, which is trained on limited number of tasks from the target environment performs close to this ideal reference, and that it coincides with it when the deviation in task distributions is zero, i.e, when source and target task environments are the same.

In Fig. 3.2, we compare the posterior predictions of the three schemes introduced above, along with GP, against the ground truth regression function. We set $\sigma = 0.1$, $\alpha = \beta = 0.2$, $\mu_c' - \mu_c = 0.5$, number of tasks $N = 30$ and number of samples per task $M_i = 5$. We adopt the MAP approximation for all learning schemes. The dashed line represents the ground truth regression line in (3.17). The performance of WFEM-GP is again comparable to the best achievable performance of a meta-learner trained on N tasks from the target environment.

In Fig. 3.3, we compare the performance of the four schemes outlined above as a function of the fraction β of tasks from the source task environment. We set $\sigma = 0.1$, $\alpha = \beta$, $\mu_c' - \mu_c = 0.75$, $N = 30$ and $M_i = 5$. When $\beta = 0$, only tasks from the target environment are available for meta-training, and hence the PACOH-GP and WFEM-GP schemes coincide. At the other extreme, when $\beta = 1$, i.e., only tasks from the source environment are available for meta-training, PACOH-GP using $(1-\beta)N$ target tasks coincides with GP as the two schemes share the same initial hyperparameter vector $\boldsymbol{\theta}$. In

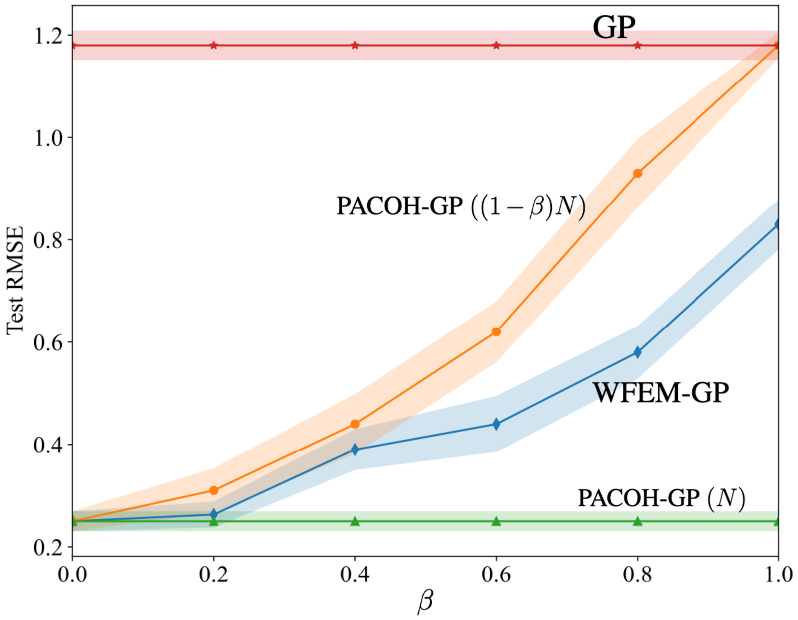

Figure 3.3 Average test RMSE under four schemes – GP, PACOH-GP with N tasks and with $(1-\beta)N$ tasks from the target environment and WFEM-GP – as a function of β with fixed $\mu_c' - \mu_c = 0.75$.

general, as β increases, WFEM-GP increasingly deviates from the ideal performance of the meta-learner trained on N target tasks, while clearly outperforming PACOH-GP with $(1-\beta)N$ target tasks.

In Fig. 3.4, we also investigate the impact of varying the weight parameter α on the performance of WFEM-GP in the regression experiment. We set $\sigma = 0.1$, $\beta = 0.4$, $\mu_c' - \mu_c = 0.75$, $N = 30$ and $M_i = 5$. Tuning the weighing parameter α is seen to be important to optimize the accuracy. For $\beta = 0.4$, the optimal performance corresponds to setting $\alpha \approx 0.2$. This indicates that one can partition data samples from each task and perform multiple rounds of cross-validation to select the optimal weighing parameter α in practice [165].

3.6 Conclusion

In this work, we have introduced WFEM, a novel transfer meta-learning approach that leverages data from both meta-training and meta-test environments. And we specifically focus on GP as the base learning model to demonstrate the performance improvement on a regression problem. The key mechanism underlying WFEM-GP involves generalizing IMRM for transfer meta-learning via optimizing the free energy objective on the distribution over shared inter-task variables, which are updated following Bayesian principles. The proposed WFEM-GP incurs the same computational complexity as the state-of-the-art PACOH-GP while providing a better generalization on OOD test data.

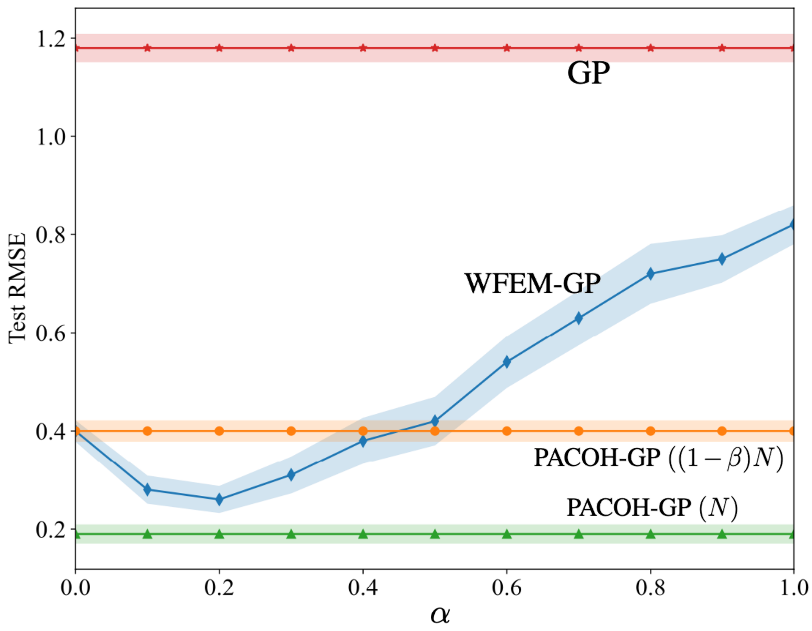

Figure 3.4 Average test RMSE under four schemes – GP, PACOH-GP with N tasks and with $(1-\beta)N$ tasks from the target environment and WFME-GP – as a function of α with fixed $\mu_c' - \mu_c = 0.75$, $\beta = 0.4$, $\sigma = 0.1$, $N = 30$ and $M_i = 5$.

Future work may address how WFEM-GP can be applied to BO for a meta-learned optimization strategy where the optimization task distribution shifts over time. Other possible extensions include replacing the GP with Bayesian neural network (BNN) for better scalability [35], as well as enabling scalable BO [160].

Chapter 4

Bayesian and Multi-Armed Contextual Meta-Optimization for Efficient Wireless Radio Resource Management

4.1 Overview

In this Chapter, we turn to the application of meta-learning to BO on a wireless communication problem. Optimal resource allocation in modern communication networks calls for the optimization of objective functions that are only accessible via costly separate evaluations for each candidate solution. The conventional approach carries out the optimization of resource-allocation parameters for each system configuration, characterized, e.g., by topology and traffic statistics, using global search methods such as BO. These methods tend to require a large number of iterations, and hence a large number of key performance indicator (KPI) evaluations. In this paper, we propose the use of meta-learning to transfer knowledge from data collected from related, but distinct, configurations in order to speed up optimization on new network configurations. Specifically, we combine meta-learning with BO, as well as with multi-armed bandit (MAB) optimization, with the latter having the potential advantage of operating directly on a discrete search space. Furthermore, we introduce novel contextual meta-BO and meta-MAB algorithms, in which transfer of knowledge across configurations occurs at the level of a mapping from graph-based contextual information to resource-allocation parameters. Experiments for the problem of open loop power control (OLPC) parameter optimization for the uplink of multi-cell multi-antenna systems provide insights into the potential benefits of meta-learning and contextual optimization.

4.2 Introduction

4.2.1 Context and Scope

The management and configuration of modern cellular communication systems requires the optimization of a large number of parameters that define the operation across all segments of the network, including the radio access network (RAN) [140]. Machine learning, or artificial intelligence (AI), methods are often invoked as potential solutions, and most efforts in this direction leverage neural network-based methods, which may incorporate contextual information such as on the network topology [21, 155, 59, 170]. However, the implementation of AI solutions for resource allocation is practically constrained by the limited access of the designer to relevant data and to efficiently computable objective functions. In fact, typically, each candidate solution can only be evaluated via a point-wise estimate of *key performance indicators (KPIs)* through expensive simulations or measurements [81]. This paper investigates methods that aim at reducing the number of KPI evaluations needed for AI-based resource allocation via the introduction of novel optimizers based on meta-learning [22, 157], multi-armed bandit optimization [50], and contextual optimization [88].

To exemplify the application of the proposed resource-allocation optimizers, we focus on the important problem of *open loop power control (OLPC)* for the uplink of a multi-cell system with multi-antenna base stations [112] (see Fig. 4.1). This optimization requires a search over a large discrete space of candidate options, and each candidate power control parameter set needs to be evaluated via the use of a network simulator or via measurements in the field. The conventional approach carries out the optimization of resource-allocation parameters for each system configuration, which is characterized, e.g., by topology and traffic statistics [209]. This *per-configuration* approach is justified by the diversity of network deployments, which generally prevents the direct reuse of solutions found for one deployment to another deployment. However, as mentioned, this class of solutions is practically impaired by the need to evaluate many candidate solutions as intermediate steps towards a satisfactory solution.

4.2.2 Related Work

Machine learning solutions based on deep neural networks (DNNs) train a generic dense neural network in a supervised or unsupervised fashion to approximate the output of model-based power control algorithms such as the Weighted Minimum Mean Squared Error (WMMSE) [169, 29, 99, 95, 90]. Alternatively, reinforcement learning can be leveraged to autonomously optimize channel selection and power allocation based on feedback from the network designer [176]. Unlike methods based on supervised or unsupervised learning, reinforcement learning does not rely on a model-based optimizer and it does not require

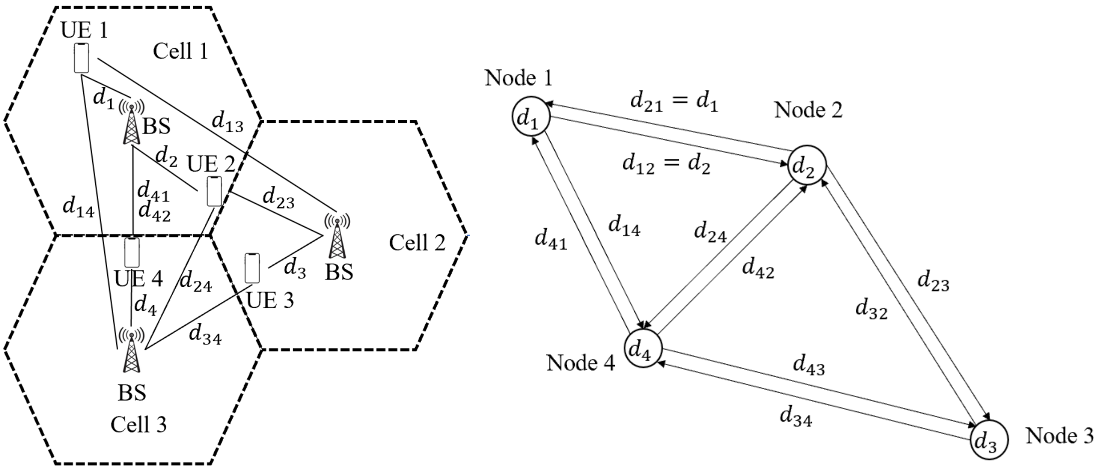

Figure 4.1 A configuration τ is described in this example by the network topology illustrated on the left. The network encompasses $N^C = 3$ cells, each with one BS. There are four UEs, with $N_{U,1} = 2$, $N_{U,2} = 1$, and $N_{U,3} = 1$ UEs in cells 1, 2, and 3, respectively. Therefore, communication links exist between UE 1 and the BS in cell 1, UE 2 and the BS in cell 1, UE 3 and the BS in cell 2, as well as UE 4 and the BS in cell 3. Meta-learning schemes based on BO or MAB optimize power allocation for this network configuration based on KPI measurement from other network configurations, characterized, e.g., by different distances or number of UEs per cell. Furthermore, as explained in Sec. VI, for contextual optimization, the context vector $\mathbf{c}_\tau$ may contain all distances, where d_i is the distance from UE-i to the serving BS and d_{ij} is the distance between UE-i and the BS serving UE-j. The context vector $\mathbf{c}_\tau$ can be described in terms of the interference graph $\mathcal{G}_\tau$ shown on the right. In the graph, each node corresponds to one of the four links, and is marked with the relevant distance between UE and serving BS. A directed edge is included between links for which the distance between the transmitting UE for the first link and the receiving BS for the second link is sufficiently small, indicating a meaningful level of interference between the first link and the second link.

access to gradients of the objective function, but it typically necessitates many evaluations of the KPIs of interest at intermediate solutions.

It was recently pointed out by some of the authors of the present contribution in [112] that BO with GP can provide a more flexible solution that does not require access to gradient information for the objective function and can potentially reduce convergence time for power control optimization as compared to reinforcement learning. However, BO still requires a separate optimization for each network configuration, and the number of per-configuration KPI evaluations may still be prohibitively high.

Contextual BO was studied in [88]. In this reference, the BO optimizer is given a different context vector at each optimization step. For this situation, the authors of [88] propose to append the context vector to the input. This approach does not work well for the problem of interest in which the context vector is fixed at run time, and hence different solutions must be compared for the same context vector. This calls for the use of a distinct context-based optimization approach, which we introduce in this work.

4.2.3 Main Contributions

In this chapter, we propose for the first time the use of meta-learning to transfer knowledge from data collected from related, but distinct, network configurations in order to speed up optimization of resource allocation parameters on new network configurations. The speed-up is measured in terms of the number of evaluations of KPIs for candidate solutions that are needed to attain an effective resource allocation strategy. To this end, our contributions are of both methodological and application-based nature. Specifically, we introduce new meta-learning-based design methodologies, which we expect to be of independent interest and broader applicability; and we investigate their application to uplink OLPC in cellular systems. The proposed methods leverage the availability of offline data from multiple network configurations, or deployments, to tailor OLPC adaptation strategies for any new deployment.

The main contributions of this chapter are as follows:

- At a methodological level, we introduce a novel scheme that combines meta-learning with *multi-armed bandit (MAB) optimization* [50]. MAB has the potential advantage over BO of operating directly on a discrete search space. This is a particularly useful feature in problems, such as OLPC, in which the optimization variables are quantized. Our approach, termed *meta-MAB*, is based on a specific parameterization of the *Exponential-weight algorithm for Exploration and Exploitation (Exp3)* bandit selection policy [110] that enables meta-optimization based on data from multiple tasks.

- Also at a methodological level, we propose novel *contextual* meta-BO and meta-MAB algorithms that can incorporate task-specific information in the form of a graph. The proposed approach is based on a graph kernel formulation [196], whereby problems characterized by similar contextual graph information are assigned related solutions. In the context of the OLPC problem, contextual meta-BO and meta-MAP optimize a mapping from graph-based contextual information about the network topology to power allocation parameters (see Fig. 4.1).

- In terms of applications, we propose for the first time to leverage meta-BO and meta-MAB for optimal resource allocation with a focus on the problem of OLPC parameter optimization. As mentioned, while meta-BO is directly applicable to continuous search spaces, and can also be adapted to work for discrete optimization, meta-MAB directly targets discrete search spaces. The benefit of the proposed meta-BO and meta-MAB strategies is the reduction in the number of KPI evaluations, or iterations, needed to optimize resource allocation for each new configuration.

- We validate the performance of all the proposed methods in a multi-cell system following 3rd Generation Partnership Project (3GPP) specifications. Experiments for the problem of OLPC parameter optimization provide insights into the potential benefits of meta-learning and contextual optimization strategies.

The rest of this chapter is organized as follows. First, in Sec. 4.3 we formulate the problem. Sec. 4.4 introduces meta-BO; while Sec. 4.5 reviews MAB and proposed meta-MAB. Contextual meta-BO and meta-MAB are introduced in Sec. 4.6, and experimental results are provided in Sec. 4.7. Sec. 4.8 concludes the paper.

4.3 Problem Formulation

We consider the problem of uplink power allocation in a wireless cellular communication system with N_C cells, with each cth cell containing one multi-antenna base station (BS) and $N_{U,c}$ user equipments (UEs). As in [112], we specifically focus on the optimization of long-term uplink power control parameters that are network-controlled and updated infrequently by the network operator. Accordingly, the power-control parameters are not adapted in real time, i.e., at time scale of milliseconds, but rather at the scale of hours – e.g., peak vs. non-peak times – or days – e.g., weekday vs. week-end.

In each cell c, the BS is equipped with $N_{R,c}$ receiving antennas, and each UE u has $N_{T,c,u}$ transmit antennas. Note that different UEs, such as smart watches, smart phones, or sensors, generally have a distinct number of antennas, which may not be known at the network side. Let $\mathrm{P}_{\mathbf{H}}$ denote the probability distribution of the instantaneous channel state information (CSI) $\mathbf{H}$ describing the propagation channels between the BSs and all the UEs. The channel distribution $\mathrm{P}_{\mathbf{H}}$ may account for the environment type, e.g., rural, urban, or industrial; for the locations of the UEs and BSs; as well as for slow and fast fading effects, including blockages. The user activity can be also implicitly modelled by the distribution $\mathrm{P}_{\mathbf{H}}$, as inactive UEs can be modelled as having negligible connectivity to all BSs.

We define the *configuration* τ of the system via the tuple $\tau = (\mathbf{N}_R, \mathbf{N}_U, \mathbf{N}_T, \mathrm{P}_{\mathbf{H}})$ consisting of vectors $\mathbf{N}_R$ and $\mathbf{N}_T$, which collect the numbers of antennas at BSs and UEs across the cells, respectively; of vector $\mathbf{N}_U$, which counts the number of UEs in each cell, and of the CSI distribution $\mathrm{P}_{\mathbf{H}}$. We are interested in developing efficient solutions for power allocation of the UEs given any system configuration τ. We first focus on developing efficient solutions for power allocation of the UEs given any system configuration τ. Then, in Sec. 4.6, we consider a more general setting in which the power control policy can also depend on "context" information about the CSI distribution $\mathrm{P}_{\mathbf{H}}$, such as the topology of the network.

For a given configuration τ, the distribution $\mathrm{P}_{\mathbf{H}}$ is generally unknown. For instance, the UE distribution and/or fading models may not be available. Power control can be based only on the vectors $\mathbf{N}_R, \mathbf{N}_U, \mathbf{N}_T$, as well as on a dataset $\mathcal{D}_\tau = \{\mathbf{H}_{\tau,s}\}_{s=1}^{S_\tau}$ of S_τ CSI realizations. The dataset $\mathcal{D}_\tau$ is practically obtained through channel estimation procedures. Our goal is to design mechanisms that can optimize the power allocation strategy for any

Table 4.1 Allowed values for OLPC parameters

P_0 (dBm)	$-202, -200, ..., +22, +24$
α	$0, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$

new configuration τ even when only few data points are available, i.e., when S_τ is small, and/or when limited time and computational power can be expended for optimization. To this end, we will combine an offline meta-optimization step with an adaptation step based on dataset $\mathcal{D}_\tau$. In practice, as we will discuss, one may not have access to CSI, but only to point-wise measurements of a relevant *key performance indicator (KPI)*, and the aim is to minimize the number of such measurements required to identify a well performing power control solution.

According to the 3GPP's fractional power control policy [19], each UE u in cell c calculates its transmitting power $P^{\text{TX}}_{c,u}$ (in dBm) on the *physical uplink shared channel* (PUSCH) as a function of the OLPC $(P_{0,c}, \alpha_c)$. These consist of the expected power $P_{0,c}$ received at the BS of cell c under full power compensation, and the fractional power control compensation parameter $\alpha_c \in [0,1]$ for cell c. Specifically, focusing on a single resource block, the power $P^{\text{TX}}_{c,u}$ is obtained as [19]

$$P^{\text{TX}}_{c,u} = \min\{P^{\max}_{c,u}, P_{0,c} + \alpha_c \text{PL}_{c,u} + \text{CL}_{c,u}\} \quad [\text{dBm}], \tag{4.1}$$

where $P^{\max}_{c,u}$ is the maximum UE transmit power; and $\text{PL}_{c,u}$ is the pathloss in dB towards the serving cth BS, and $\text{CL}_{c,u}$ is the closed-loop power control adjustment for UE u. Note that, by (4.1), if $\alpha_c = 1$ the received power is $P_{0,c} + \text{CL}_{c,u}$, unless the maximum power constraint $P^{\max}_{c,u}$ forces the equality $P^{\text{TX}}_{c,u} = P^{\max}_{c,u}$ in (4.1). The OLPC parameters $(P_{0,c}, \alpha_c)$ are generally distinct across the cells, i.e., they depend on the cell index c. Furthermore, they are constrained to lie in the set of $N_{OLPC} = 912$ options described in Table 4.1 [112]. We define as $\mathbf{P}_0 = [P_{0,1}, \ldots, P_{0,N_C}]^T$ the $N_C \times 1$ vector of expected received power parameters across all cells; and as $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_{N_C}]^T$ as the vector of fractional power compensation parameters. Note that the optimization space, i.e., the number of allowed values of the OLPC parameters P_0 and α grows exponentially with the number of cells N_C.

The OLPC parameters $(\mathbf{P}_0, \boldsymbol{\alpha})$ are to be selected so as to optimize a given uplink KPI [112]. The KPI obtained for a given CSI $\mathbf{H}_\tau$ is a function of the OLPC parameters $(\mathbf{P}_0, \boldsymbol{\alpha})$ through (4.1), and is denoted as KPI$(\mathbf{P}_0, \boldsymbol{\alpha}, \mathbf{H}_\tau)$. The KPI may be obtained via fixed measurements or through the use of a simulator. For any given configuration τ, we are interested in maximizing the average network-wide KPI as per the discrete optimization problem

$$\max_{\mathbf{P_0}, \boldsymbol{\alpha}} \left\{ \mathbb{E}_{P_{\mathbf{H}_\tau}} \left[\text{KPI}(\mathbf{P}_0, \boldsymbol{\alpha}, \mathbf{H}_\tau) \right] \right\}, \tag{4.2}$$

Algorithm 1: Bayesian Optimization (BO) for a given configuration τ

```
Input: GP prior $(\mu(\cdot), k(\cdot,\cdot))$, CSI dataset $\mathcal{D}_\tau$, maximum number of rounds $T_{max}$
Output: Optimized $\mathbf{x}^*$
Initialize round $t = 0$, empty matrix $\mathbf{X}_0 = [\,]$, empty vector $\tilde{\mathbf{f}}_0 = [\,]$
while not converged do
    Obtain the next OLPC vector $\mathbf{x}_{t+1}$ using (2.11)
    Obtain observation $\tilde{f}_{t+1} \sim \mathcal{N}(\tilde{f}_{t+1} | f(\mathbf{x}_{t+1}), \sigma^2)$
    Update matrix $\mathbf{X}_{t+1} = [\mathbf{X}_t, \mathbf{x}_{t+1}]$ and vector $\tilde{\mathbf{f}}_{t+1} = [\tilde{\mathbf{f}}_t, \tilde{f}_{t+1}]^\mathsf{T}$
    Set $t = t + 1$
end
Return $\mathbf{x}^* = \mathbf{x}_{t^*}$ with $t^* = \arg\max_{t' \in \{1,\ldots,t-1\}} \tilde{f}_{t'}$
```

where the objective function in (4.2) is expressed as the average KPI over the CSI distribution $\mathrm{P}_{\mathbf{H}_\tau}$ for configuration τ. Examples of KPI include the sum-achievable rate, as it will be detailed in Sec. 4.7.

Intuitively, if α_c and $P_{0,c}$ are large, the intended received power at the BS of cell c is high, but the interference generated to neighboring BSs is also significant. Conversely, if α_c and $P_{0,c}$ are small, both intended signal and interference are low. Therefore, the solution of problem (4.2) hinges on the identification of an optimized trade-off between intra-cell received power and inter-cell interference.

The objective in (4.2) cannot be directly evaluated, since it depends on the unknown distribution $\mathrm{P}_{\mathbf{H}_\tau}$. However, it can be estimated by using the CSI dataset $\mathcal{D}_\tau$ via the empirical average

$$f_\tau(\mathbf{P}_0, \boldsymbol{\alpha}) = \frac{1}{S_\tau} \sum_{s=1}^{S_\tau} \mathrm{KPI}(\mathbf{P}_0, \boldsymbol{\alpha}, \mathbf{H}_{\tau,s}), \tag{4.3}$$

where we recall that S_τ is the number of available measurements $\{\mathbf{H}_{\tau,s}\}$ in dataset $\mathcal{D}_\tau$. Overall, the problem of interest is the optimization

$$\max_{\mathbf{P_0}, \boldsymbol{\alpha}} f_\tau(\mathbf{P}_0, \boldsymbol{\alpha}). \tag{4.4}$$

When one restricts the parameters $(\mathbf{P}_0, \boldsymbol{\alpha})$ as in Table 4.1, the problem is discrete.

One could solve the discrete optimization problem (4.4) using exhaustive search, but this may not be computationally feasible. In fact, the optimization space includes N_{OLPC} possible OLPC choices. In the next sections, we will explore more efficient, approximate solutions. As we will detail in Sec. 4.4, the proposed meta-learning methods leverage the principle of transferring knowledge from previously encountered configurations τ in order to prepare to optimize power allocation for new configurations.

4.4 Bayesian Meta-Optimization

As reviewed in Chapter 2, we construct the GP regression and perform conventional BO summarized in Algorithm 1 by treating $\mathbf{x} = [\mathbf{P}_0^\mathsf{T}, \boldsymbol{\alpha}^\mathsf{T}]^\mathsf{T}$ for the vector of variables under optimization in problem (4.4). Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T]$ denote any set of T inputs, $\mathbf{f}(\mathbf{X}) = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_T)]^\mathsf{T}$ denote the corresponding KPI values, and the vector of noisy observed KPI values is denoted as $\tilde{\mathbf{f}} = [\tilde{f}_1, ..., \tilde{f}_T]^\mathsf{T}$. Accordingly, the GP posterior distribution can be built as

$$p(f(\mathbf{x}) = f|\mathbf{X}, \tilde{\mathbf{f}}) = \mathcal{N}(f|\mu(\mathbf{x}|\mathbf{X}, \tilde{\mathbf{f}}), \sigma^2(\mathbf{x}|\mathbf{X}, \tilde{\mathbf{f}})) \tag{4.5}$$

with mean and variance function following (2.7) and (2.8) with the corresponding notations in this section.

Solving problem (4.4) separately for each configuration τ via Bayesian optimization (Algorithm 1) may entail significant complexity in terms of number S_τ of required CSI samples, as well as number of evaluations of KPI values, i.e., the number of iterations in Algorithm 1. In this section, we introduce Bayesian meta-optimization [130, 69], which uses offline data collected from multiple system configurations τ as a means to reduce optimization complexity when applied to any configuration τ at run time.

In Bayesian meta-optimization, we assume that, in an offline phase, we can collect data from N configurations, denoted as $\tau_1, ..., \tau_N$. These configurations may correspond to previous deployments or to concurrent deployments located elsewhere the system or to previous runs of a simulator with different settings, such as inter-site distances and number of UEs. For each configuration τ_n, with $n = 1, ..., N$, we have access to a dataset $\mathcal{D}_{\tau_n}$ of S_{τ_n} CSI samples, which can be used to obtain the objective function $f_{\tau_n}(\mathbf{x})$ in (4.3). Furthermore, for each task τ_n, we assume to have collected T_n inputs $\mathbf{X}_n = [\mathbf{x}_{n,1}, ..., \mathbf{x}_{n,T_n}]$, as well as the corresponding noisy observations $\tilde{\mathbf{f}}_n = [\tilde{f}_{n,1}, ..., \tilde{f}_{n,T_n}]$ of the actual objective values $f_{n,t} = f_{\tau_n}(\mathbf{x}_{n,t})$. We refer to the above collected data available from N configurations as *meta-training data*. In practice, the designer may equivalently only have access to T_n evaluations of the KPI function. In our experiments, we explore values T_n in the range $[1, 30]$. We aim at using these data to improve efficiency on new tasks sampled from the same environment.

To this end, Bayesian meta-optimization uses meta-training data to optimize the GP prior via parametric mean function $\mu_{\boldsymbol{\theta}}(\cdot)$ and kernel function $k_{\boldsymbol{\theta}}(\cdot)$, which are functions of a vector of *hyperparameters* $\boldsymbol{\theta}$. Specifically, we consider the parametric kernel function [144]

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \exp(-||\psi_{\boldsymbol{\theta}}(\mathbf{x}) - \psi_{\boldsymbol{\theta}}(\mathbf{x}')||_2^2), \tag{4.6}$$

where $\psi_{\boldsymbol{\theta}}(\cdot)$ is a neural network with hyperparameter vector $\boldsymbol{\theta} \in \mathbb{R}^L$ constituting its synaptic weights and biases and we also assume $\mu_{\boldsymbol{\theta}}(\mathbf{x})$ to be a neural network. By optimizing the GP prior via (4.6), the goal is to ensure that Bayesian optimization applied

Algorithm 2: Bayesian Meta-Optimization (Meta-BO)

```
Input : Parameterized GP prior $(\mu_{\boldsymbol{\theta}}(\cdot), k_{\boldsymbol{\theta}}(\cdot,\cdot))$, meta-training data $\mathbf{X}_{1:N}$, $\tilde{\mathbf{f}}_{1:N}$,
        stepsize $\beta$
Output : Optimized hyperparameters vector $\boldsymbol{\theta}^*$
Initialize hyperparameters vector $\boldsymbol{\theta}$
while not done do
    Evaluate gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}|\mathbf{X}_{1:N}, \tilde{\mathbf{f}}_{1:N})$ using (4.10)
    Update hyper-parameters using gradient descent $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}|\mathbf{X}_{1:N}, \tilde{\mathbf{f}}_{1:N})$
end
Return $\boldsymbol{\theta}^*$
Given a new network configuration $\tau$, apply BO with hyperparameter $\boldsymbol{\theta}^*$
```

to a new configuration τ can produce an effective solution with fewer samples S_τ and fewer evaluations T_{max} of the KPI.

Intuitively, the role of the kernel function is to quantify the similarity between power control parameters x and x' in terms of the respective KPI values obtained for a given configuration. The standard approach in BO is to select this kernel as a predefined distance metric, e.g., the Euclidean distance in [112], which may not reflect well the specific properties of the given optimization problem (4.4). In contrast, Bayesian meta-optimization aims at optimizing the kernel function so as to account for the structure of the power control optimization problems (4.4) for the N configurations for which we have meta-training data. The rationale is that one expects such structure to be sufficiently related to that of any new configuration τ of interest.

Bayesian meta-optimization, is formulated by introducing *meta-training loss* incurred on the meta-training data $\mathbf{X}_{1:N} = [\mathbf{X}_1, ..., \mathbf{X}_N]$ and $\tilde{\mathbf{f}}_{1:N} = [\tilde{\mathbf{f}}_1, ..., \tilde{\mathbf{f}}_N]$ when using hyperparameter vector $\boldsymbol{\theta}$ as

$$\mathcal{L}(\boldsymbol{\theta}|\mathbf{X}_{1:N}, \tilde{\mathbf{f}}_{1:N}) = -\frac{1}{N}\sum_{n=1}^{N}\frac{1}{T_n}\ln p_{\boldsymbol{\theta}}(\tilde{\mathbf{f}}_n|\mathbf{X}_n), \tag{4.7}$$

where

$$\begin{aligned}\ln p_{\boldsymbol{\theta}}(\tilde{\mathbf{f}}_n|\mathbf{X}_n) = &-\frac{1}{2}\Big(\tilde{\mathbf{f}}_n - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{X}_n)\Big)^{\mathsf{T}}\Big(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_n)\Big)^{-1}\Big(\tilde{\mathbf{f}}_n - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{X}_n)\Big)\\ &-\frac{1}{2}\ln\Big|\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_n)\Big| - \frac{T_n}{2}\ln(2\pi),\end{aligned} \tag{4.8}$$

with $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{X}_n) = [\mu_{\boldsymbol{\theta}}(\mathrm{x}_{n,1}), ..., \mu_{\boldsymbol{\theta}}(\mathrm{x}_{n,T_n})]^{\mathsf{T}}$; $[\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}_n)]_{t,t'} = k_{\boldsymbol{\theta}}(\mathrm{x}_{n,t}, \mathrm{x}_{n,t'})$ for $(t,t') \in \{1, ..., T_n\}$; and $\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_n) = \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}_n) + \sigma^2\mathbf{I}_{T_n}$. The meta-training loss (4.7) is the empirical average of the negative log-likelihood evaluated on the meta-training data [144]. Alternatively, one may adopt importance sampling methods [43, 166] to mitigate the non-i.i.d. meta-training data induced by the acquisition function, e.g., less weight assigned to similar power control parameters in $\mathbf{X}_n$. The optimal hyperparameter $\boldsymbol{\theta}^*$ is obtained by addressing the problem

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}|\mathbf{X}_{1:N}, \tilde{\mathbf{f}}_{1:N}). \tag{4.9}$$

To implement the optimization in (4.9), we adopt a gradient-based optimizer. The partial derivative of the meta-training loss with respect to the j-th component θ_j of the hyperparameters vector $\boldsymbol{\theta}$ is computed as

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\mathcal{L}(\boldsymbol{\theta}|\mathbf{X}_{1:N}, \tilde{\mathbf{f}}_{1:N}) = -\frac{1}{N}\sum_{n=1}^{N}\Bigg(&\frac{1}{2}\Big(\tilde{\mathbf{f}}_n - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{X}_n)\Big)^{\mathsf{T}}\Big(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_n)\Big)^{-1} \\ &\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_n)}{\partial\theta_j}\Big(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_n)\Big)^{-1}\Big(\tilde{\mathbf{f}}_n - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{X}_n)\Big) \\ &-\frac{1}{2}\mathrm{tr}\bigg(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_n)^{-1}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_n)}{\partial\theta_j}\bigg)\Bigg)\frac{1}{T_n} \\ = -\frac{1}{N}\sum_{n=1}^{N}\frac{1}{2T_n}\mathrm{tr}\bigg(&\Big(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^{\mathsf{T}} - \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_n)^{-1}\Big)\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_n)}{\partial\theta_j}\bigg),\end{aligned} \tag{4.10}$$

where $\boldsymbol{\Lambda} = \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_n)^{-1}(\tilde{\mathbf{f}}_n - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{X}_n))$. The partial derivative term in (4.10) can be estimated by backprop with the parameters in (4.6).

The hyper-parameter $\boldsymbol{\theta}^*$ optimized with the gradient-based procedure outlined above is used to define the GP prior to be used for Bayesian optimization in new configurations for the purpose of improving the efficiency of Bayesian optimization. Overall, Bayesian meta-optimization is summarized in Algorithm 2.

4.5 Bandit Optimization and Meta-optimization

Given the discrete nature of problem (4.4) when considering Table 4.1, it can be directly modelled as a stochastic multi-armed bandit (MAB) model rather than as GP, which assumes continuous variables. In the MAB formulation, the total number of arms equals the number, N_{OLPC}, of OLPC parameters options listed in Table 4.1. The goal is to design a policy that selects the best "arm" i.e., the OLPC pair $(\mathbf{P}_0, \boldsymbol{\alpha})$ that optimizes problem (4.4) after a small number of attempts. In practice, as in the case of Bayesian optimization, one accepts sub-optimal solutions that performs well enough.

4.5.1 Bandit Policy

As in Bayesian optimization (see Algorithm 1), for a configuration τ, at the tth optimization round, the learning agent selects an OLPC configuration $\mathbf{x}_t$ from Table 4.1 and observes a noisy version $\tilde{f}_t$ of the corresponding KPI value $f_\tau(\mathbf{x}_t)$. In a manner similar to (2.11), a bandit optimization policy maps the history $(\mathbf{X}_t, \tilde{\mathbf{f}}_t)$ of previous selections and

Algorithm 3: Multi-Armed Bandit Optimization (MAB) for a given configuration τ

```
Input : Policy parameter $\omega$, CSI dataset $\mathcal{D}_\tau$, maximum number of rounds $T_{max}$
Output : Optimized $\mathbf{x}^*$
Initialize round $t = 0$, empty matrix $\mathbf{X}_0 = [\,]$, empty vector $\tilde{\mathbf{f}}_0 = [\,]$
while not converged do
    Sample from policy $p_\omega(\mathbf{x}|\mathbf{X}_t, \tilde{\mathbf{f}}_t)$ to obtain $\mathbf{x}_{t+1}$
    Obtain observation $\tilde{f}_{t+1} \sim \mathcal{N}(\tilde{f}_{t+1}|f(\mathbf{x}_{t+1}), \sigma^2)$
    Update matrix $\mathbf{X}_{t+1} = [\mathbf{X}_t, \mathbf{x}_{t+1}]$ and vector $\tilde{\mathbf{f}}_{t+1} = [\tilde{\mathbf{f}}_t, \tilde{f}_{t+1}]^\mathsf{T}$
    Set $t = t+1$
end
Return $\mathbf{x}^* = \arg\max_{t' \in \{1,\dots,t-1\}} \tilde{f}_{t'}$
```

corresponding cost functions up to round t to the next selection $\mathbf{x}_{t+1}$. Specifically, we consider a stochastic bandit policy $p_\omega(\mathbf{x}|\mathbf{X}_t, \tilde{\mathbf{f}}_t)$, parameterized by a scalar $\omega \in [0,1]$, that defines the probability of selecting an OLPC configuration $\mathbf{x}$ at t-th round given the past history $(\mathbf{X}_t, \tilde{\mathbf{f}}_t)$. Policy $p_\omega(\mathbf{x}|\mathbf{X}_t, \tilde{\mathbf{f}}_t)$ can be defined via a recurrent neural network [16] or via simpler functions such as the Exp3 policy in [110].

In this work, we consider the following *modified Exp3 policy*

$$p_\omega(\mathbf{x}|\mathbf{X}_t, \tilde{\mathbf{f}}_t) = (1-\omega)\frac{\exp(G(\mathbf{x}, t-1))}{\sum_{\mathbf{x}'} G(\mathbf{x}', t-1)} + \frac{\omega}{N_{OLPC}}, \tag{4.11}$$

where $\omega \in [0,1]$ is the policy parameter; the sum is over all the possible OLPC configurations in Table 4.1; and

$$G(\mathbf{x}, t-1) = \sum_{i=1}^{t-1} k(\mathbf{x}_i, \mathbf{x})[p_\omega(\mathbf{x}|\mathbf{X}_{t-1}, \tilde{\mathbf{f}}_{t-1})]^{-1}\tilde{f}_i, \tag{4.12}$$

is a weighted average of the noisy objective function values obtained for input $\mathbf{x}$ in the previous $t-1$ rounds, with $k(\cdot,\cdot)$ being a kernel function. While the conventional choice for the kernel function is the identity function $k(\mathbf{x}, \mathbf{x}') = 1$ if $\mathbf{x} = \mathbf{x}'$ and $k(\mathbf{x}, \mathbf{x}') = 0$ otherwise, here we will allow for a more general solution. This will be useful in the next subsection to facilitate the application of meta-learning.

Standard bandit optimization considers a fixed parameter parameter ω, and is summarized in Algorithm 3.

4.5.2 Bandit Meta-Optimization

Following the meta-learning setting introduced in Sec. 4.4, in this section we propose a bandit meta-optimization strategy. As in Sec. 4.4, we assume availability of data for N system configurations. The goal of bandit meta-optimization is to use such meta-

Algorithm 4: Bandit Meta-Optimization (Meta-MAB)

```
Input : Parameterized policy $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{X}_n, \tilde{\mathbf{f}}_n)$, meta-training data $\mathbf{X}_{1:N}$, $\tilde{\mathbf{f}}_{1:N}$, stepsize $\eta$
Output : Optimized policy vector $\boldsymbol{\theta}^* = (\boldsymbol{\varphi}^*, \omega^*)$
Initialize policy vector $\boldsymbol{\theta} = (\boldsymbol{\varphi}, \omega)$
while not done do
    Evaluate gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}|\mathbf{X}_{1:N}, \tilde{\mathbf{f}}_{1:N})$ using (4.14)
    Update policy vector using gradient descent $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}|\mathbf{X}_{1:N}, \tilde{\mathbf{f}}_{1:N})$;
end
Return $\boldsymbol{\theta}^*$
Given a new network configuration $\tau$, apply MAB (Algorithm 3) with hyperparameter $\boldsymbol{\theta}^* = (\boldsymbol{\varphi}^*, \omega^*)$ with kernel function $k_{\boldsymbol{\varphi}}(\cdot,\cdot)$
```

training data, given by $\mathbf{X}_{1:N}$ and $\tilde{\mathbf{f}}_{1:N}$ as defined in the previous sections, to optimize a hyperparameter vector defining the bandit policy.

To this end, we propose to instantiate the kernel function $k_{\boldsymbol{\varphi}}(\cdot,\cdot)$ in the Exp3 policy (4.11) as in (4.6) with neural network parameters $\boldsymbol{\varphi}$. We aim at optimizing the parameter tuple $\boldsymbol{\theta} = (\boldsymbol{\varphi}, \omega)$ defining the resulting policy $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{X}_n, \tilde{\mathbf{f}}_n)$ to ensure that bandit meta-optimization applied to a new configuration τ can select an effective OLPC vector with a smaller number of trials.

To this end, we define the following meta-training loss as

$$\mathcal{L}(\boldsymbol{\theta}|\mathbf{X}_{1:N}, \tilde{\mathbf{f}}_{1:N}) = -\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\theta}}(\cdot|\mathbf{X}_n,\tilde{\mathbf{f}}_n)}[\tilde{f}(\mathbf{x})], \tag{4.13}$$

where the expectation is taken with respect to the bandit policy $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{X}_n, \tilde{\mathbf{f}}_n)$ based on the available history $(\mathbf{X}_n, \tilde{\mathbf{f}}_n)$ of observations for each n-th configuration τ_n. To implement the optimization over (4.13), we adopt a gradient-based optimizer. The gradient of the meta-training loss with respect to policy vector $\boldsymbol{\varphi}$ and ω is evaluated as [16]

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}|\mathbf{X}_{1:N}, \tilde{\mathbf{f}}_{1:N}) = -\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\theta}}(\cdot|\mathbf{X}_n,\tilde{\mathbf{f}}_n)}\left[\tilde{f}(\mathbf{x})\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{X}_n, \tilde{\mathbf{f}}_n)\right]. \tag{4.14}$$

The meta-learned optimal policy vector $\boldsymbol{\theta}^* = (\boldsymbol{\varphi}^*, \omega^*)$ is then used in the bandit policy used in Algorithm 3 to optimize the OLPC variables for a new configuration. Bandit meta-optimization is summarized in Algorithm 4.

4.6 Contextual Bayesian and Bandit Meta-Optimization

In the previous sections, we have assumed that no information is available about the current configuration τ apart from the CSI dataset $\mathcal{D}_\tau$. In practice, the system may have access to *context* information about the deployment underlying the configuration τ, such as the geometric layout, expected UE positions, or the fading statistics. In this section, we introduce a generalization of the meta-optimization strategies described in Sec. 4.4 and Sec. 4.5 that can leverage configuration-specific context information to optimize OLPC parameters $\mathbf{x} = (\mathbf{P}_0, \boldsymbol{\alpha})$.

4.6.1 Context-Based Meta-Optimization

Let $\mathbf{c}_\tau$ denote a context vector specific to configuration τ, which includes all the information available at the optimizer about configuration τ. The key idea of the proposed methods is to use meta-training data from multiple tasks in order to optimize a procedure that can adapt the parameters $\boldsymbol{\theta}$ for BO or MAB optimization to the configuration-specific context $\mathbf{c}_\tau$.

Formally, for each meta-training configuration τ_n, we have access to data $(\mathbf{X}_n, \tilde{\mathbf{f}}_n, \mathbf{c}_n)$, where $\mathbf{c}_n$ is the context vector for the meta-training task τ_n. Therefore, as compared to the meta-learning settings studied in the last two sections, here we assume the additional availability of the context vector $\mathbf{c}_n$ for each task τ_n. Accordingly, at run time, the optimizer is given context vector $\mathbf{c}_\tau$ for the current configuration τ. The goal is to effectively adapt the optimizer's parameters $\boldsymbol{\theta}$ to the context vector $\mathbf{c}_\tau$ by leveraging knowledge transferred from the meta-learning tasks.

The proposed approach leverages meta-learning data to optimize a parametric mapping $q_{\mathbf{V}}(\cdot)$ between context $\mathbf{c}_\tau$ and parameters $\boldsymbol{\theta}$. The mapping depends on a parameter matrix $\mathbf{V}$ that is to be optimized based on meta-training data. Once vector $\mathbf{V}$, and hence also the parametric mapping $q_{\mathbf{V}}(\cdot)$, are fixed, an optimized per-task configuration hyperparameters $\boldsymbol{\theta}^*_\tau$ is obtained as $\boldsymbol{\theta}^*_\tau = q_{\mathbf{V}}(\mathbf{c}_\tau)$ for the new task τ.

Intuitively, an effective mapping $q_{\mathbf{V}}(\cdot)$ should map similar context vectors, defining similar configurations, into similar parameter vectors. Two context vectors are similar if the respective KPIs depend in an analogous way on the parameters $(\mathbf{P}_0, \boldsymbol{\alpha})$ under optimizations. Since, as we will detail in the next subsection, the context vector typically encodes information about the topology of the network, the mapping should account for the extent to which topologies with similar characteristics call for related optimized power control parameters $\mathbf{x}$.

In order to facilitate the optimization of mapping functions with this intuitive property, we propose here to adopt the linear function

$$q_{\mathbf{V}}(\mathbf{c}) = \sum_{n=1}^{N} \kappa(\mathbf{c}, \mathbf{c}_n)\boldsymbol{\nu}_n, \tag{4.15}$$

where we have introduced the *context kernel* function $\kappa(\mathbf{c}, \mathbf{c}')$ to measure the similarity between two context vectors $\mathbf{c}$ and $\mathbf{c}'$. As detailed in the next subsection, the context kernel function is set by the optimizer to capture the desired similarity properties between two context vectors. The mapping (4.15) depends on parameter vectors $\boldsymbol{\nu}_1, ..., \boldsymbol{\nu}_N$ of the same dimension of the parameter vector $\boldsymbol{\theta}$, which we collect in the parameter matrix $\mathbf{V} = [\boldsymbol{\nu}_1, ..., \boldsymbol{\nu}_N]$ to be optimized. Finally, introducing the vector $\boldsymbol{\kappa}(\mathbf{c}) = [\kappa(\mathbf{c}, \mathbf{c}_1), ..., \kappa(\mathbf{c}, \mathbf{c}_N)]^{\mathsf{T}}$, the mapping (4.15) can be expressed as

$$q_{\mathbf{V}}(\mathbf{c}) = \mathbf{V}\boldsymbol{\kappa}(\mathbf{c}). \tag{4.16}$$

By (4.15), or (4.16), the parameter vector

$$\boldsymbol{\theta}_{\mathcal{T}}^{*} = q_{\mathbf{V}}(\mathbf{c}_{\mathcal{T}}) \tag{4.17}$$

for the test configuration $\mathcal{T}$ is modelled as a linear combination of vectors $\boldsymbol{\nu}_n$, with each vector $\boldsymbol{\nu}_n$ being weighted by the similarity $\kappa(\mathbf{c}_{\mathcal{T}}, \mathbf{c}_n)$ between context vectors $\mathbf{c}_{\mathcal{T}}$ and $\mathbf{c}_n$. Implementing the intuition detailed at the beginning of this subsection, we can view $\boldsymbol{\nu}_n$ as the parameter vector assigned to the meta-learning configuration $\mathcal{T}_n$, and the parameter vector $\boldsymbol{\theta}_{\mathcal{T}}^{*}$ in (4.17) as being closer to vectors $\boldsymbol{\nu}_n$ corresponding to more similar configurations $\mathcal{T}_n$ according to the kernel similarity measure $\kappa(\mathbf{c}_{\mathcal{T}}, \mathbf{c}_n)$.

The parameter matrix $\mathbf{V}$ has a number of entries equal to the product of the size of model parameter $\boldsymbol{\theta}$, denoted as L, and the number N of meta-learning tasks. This may be exceedingly large, causing optimization during meta-learning to possibly overfit the meta-training data yielding poor performance on the test configuration. To address this problem, we propose to factorize the $L \times N$ matrix $\mathbf{V}$ by using a low-rank decomposition into two lower-dimensionality factors. Accordingly, we write the mapping (4.17) as

$$\boldsymbol{\theta}_{\mathcal{T}}^{*} = q_{\mathbf{V}_1, \mathbf{V}_2}(\mathbf{c}_{\mathcal{T}}) = \mathbf{V}_1 \mathbf{V}_2^{\mathsf{T}} \boldsymbol{\kappa}(\mathbf{c}), \tag{4.18}$$

which depends on the parameter matrices $\mathbf{V}_1 \in \mathbb{R}^{L \times r}$ and $\mathbf{V}_2 \in \mathbb{R}^{N \times r}$ for rank $r < \min\{L, N\}$ being a hyperparameter.

4.6.2 Context Graph Kernel

The choice of the *context kernel* $\kappa(\cdot, \cdot)$ depends on the type of information included in the context vector for each configuration. In this subsection, we introduce a solution that applies to the common situation in which the context vector includes information about the topology of the network, namely all distances between BSs and UEs. This setting is selected to demonstrate the importance of leveraging the structure inherent in the context vector, along with the corresponding symmetry properties of the mapping from context vector to model parameters. This is detailed next.

For the purpose of power allocation, information about the topology of the network is important insofar as it determines the interference pattern among the links. In particular, the order in which the links are listed in the context vector $\mathbf{c}_\tau$ is not relevant. This implies that the mapping (4.18) should be invariant to permutations of the entries of the context vector. To enforce this invariance property, we adopt the framework of *graph kernels* [196].

To this end, we summarize information about topology of the network for configuration τ by means of an annotated interference graph $\mathcal{G}_\tau$ that retains information about within-cell UE-BS distances (see, e.g., [59]). As illustrated in Fig. 4.1, in the *interference graph* $\mathcal{G}_\tau$, each node represents a link between a UE and the serving BS. Each node i is annotated with distance d_i between the corresponding UE, also indexed by i as UE-i, and the serving BS. A directed edge from node i to node j is included in graph $\mathcal{G}_\tau$ if the interference from the link associated with node i to the link associated with node j is sufficiently large. To gauge the level of interference from link i to link j, we consider the distance d_{ij} between UE-i and the BS serving UE-j. If the ratio d_{ij}/d_j of this distance to the distance between UE-j and the serving BS is above some threshold, a directed edge is added between node i and j.

The context kernel $\kappa(\mathbf{c}_\tau, \mathbf{c}_{\tau'})$ is designed to measure the similarity between the graphs $\mathcal{G}_\tau$ and $\mathcal{G}_{\tau'}$ corresponding to context vectors $\mathbf{c}_\tau$ and $\mathbf{c}_{\tau'}$, respectively. There are a number of graph kernels that one can choose from for this purpose, ranging from graphlet kernels to deep graph kernels [196]. In this work, we focus on *graphlet kernels* [153], which are defined as

$$\kappa(\mathbf{c}_\tau, \mathbf{c}_{\tau'}) = \frac{\Psi(\mathcal{G}_\tau)^\mathsf{T}\Psi(\mathcal{G}_{\tau'})}{||\Psi(\mathcal{G}_\tau)||_2||\Psi(\mathcal{G}_{\tau'})||_2}, \tag{4.19}$$

where $\Psi(\mathcal{G})$ is a vector of features extracted from the graph $\mathcal{G}$. Each such feature of vector $\Psi(\mathcal{G})$ counts the number of times a certain sub-graph is contained in the graph $\mathcal{G}$. We specifically propose to consider the following feature vector

$$\Psi(\mathcal{G}) = [\Psi_1(\mathcal{G}), ..., \Psi_{N_U-1}(\mathcal{G})]^\mathsf{T}, \tag{4.20}$$

where $\Psi_i(\mathcal{G}) =$ number of nodes with in-degree equal to i. The rationale for this choice is that interference graphs with similar connectivity, as quantified by vector (4.20), should also have similar characteristics in terms of the impact of power control decisions on interference levels. Accordingly, context vectors with a large value of the kernel (4.19) are expected to have similar optimized power control parameters. Note that vector $\Psi(\mathcal{G})$ contains a number of entries equal to the number N_U of UEs minus 1, which corresponds to the number of nodes in the interference graph $\mathcal{G}$. Furthermore, the in-degree of a node is the number of incoming edges.

4.6.3 Context-Based Bayesian Meta-Optimization

To define context-based Bayesian meta-optimization, we directly modify the meta-training loss introduced in Sec. 4.4 in (4.7) for Bayesian meta-optimization as

$$\mathcal{L}(\mathbf{V}_1, \mathbf{V}_2 | \mathbf{X}_{1:N}, \tilde{\mathbf{f}}_{1:N}, \mathbf{c}_{1:N}) = -\frac{1}{N}\sum_{n=1}^{N}\frac{1}{T_n}\ln p_{q_{\mathbf{V}_1\mathbf{V}_2}(\mathbf{c}_n)}(\tilde{\mathbf{f}}_n|\mathbf{X}_n). \tag{4.21}$$

The key difference is that the meta-training loss is now a function of the two matrix factors $\mathbf{V}_1$ and $\mathbf{V}_2$, rather than being a function directly of the parameter vector $\boldsymbol{\theta}$. In fact, the parameter $\boldsymbol{\theta}$ is adapted to the context $\mathbf{c}_n$ of each task τ_n via the mapping $q_{\mathbf{V}_1\mathbf{V}_2}(\mathbf{c}_n)$. The meta-learned optimal parameter matrices $\mathbf{V}_1^*$ and $\mathbf{V}_2^*$ are obtained as the minimizer

$$(\mathbf{V}_1^*, \mathbf{V}_2^{*\mathsf{T}}) = \arg\min_{\mathbf{V}_1,\mathbf{V}_2}\mathcal{L}(\mathbf{V}_1, \mathbf{V}_2 | \mathbf{X}_{1:N}, \tilde{\mathbf{f}}_{1:N}, \mathbf{c}_{1:N}), \tag{4.22}$$

where the optimization can be addressed via gradient-descent and backprop in a manner similar to problem (4.9).

4.6.4 Context-Based Bandit Meta-Optimization

In a similar way, context-based bandit meta-learning addresses the minimization of the meta-training loss obtained by replacing in (4.13) the model parameter vector $\boldsymbol{\theta}$ with the output of $q_{\mathbf{V}_1\mathbf{V}_2}(\mathbf{c}_n)$ of the meta-trained mapping for each task τ_n. This yields the objective

$$\mathcal{L}(\mathbf{V}_1, \mathbf{V}_2 | \mathbf{X}_{1:N}, \tilde{\mathbf{f}}_{1:N}, \mathbf{c}_{1:N}) = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\mathbf{x}\sim p_{q_{\mathbf{V}_1\mathbf{V}_2}(\mathbf{c}_n)}(\cdot|\mathbf{X}_n,\tilde{\mathbf{f}}_n)}[\tilde{f}(\mathbf{x})], \tag{4.23}$$

which can be addressed via gradient descent.

4.7 Numerical Results

In this section, we present a number of experimental results with the goal of validating the potential benefits of the proposed meta-learning and contextual meta-learning methods for uplink power allocation via Bayesian and bandit optimization.

4.7.1 Setting

We consider a multi-cell *Multi-Input Multi-Output (MIMO)* system with a wrap-around radio distance model, in which N_U UEs in each cell are equipped with N_T transmit antennas each, while the BSs serving the UEs in each cell are equipped with N_R receiving antennas. Focusing on a single resource block, the CSI $\mathbf{H}_\tau$ consists of the $N_R \times N_T$

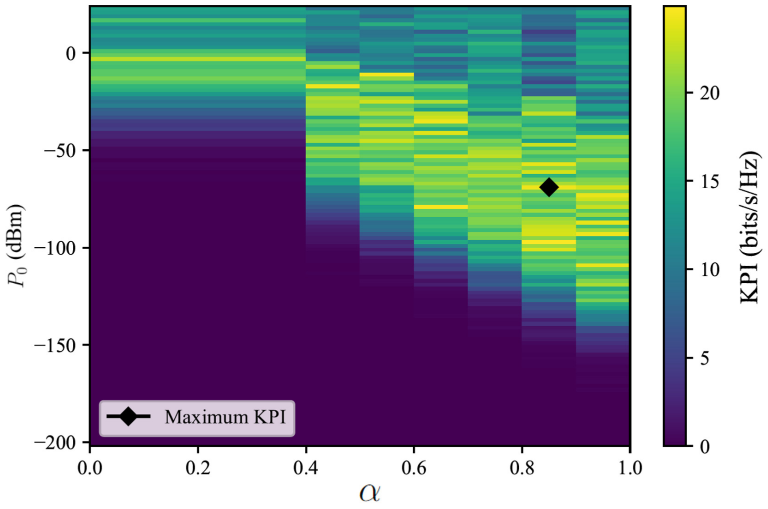

Figure 4.2 Illustration of the objective function (4.24) for a given configuration τ in the optimization space (P_0,α) for the multi-cell system considered in Sec. 4.7.

channel matrices $\mathbf{H}_{\tau,c,u,c'}$ describing the propagation channel between the N_T antennas of the uth UE in cell c and the N_R antennas of the BS in cell c'. The KPI function in (4.4) is instantiated as the sum of the spectral efficiencies for all users in the system, where the intra-cell and inter-cell signals are treated as interference. This yields (see, e.g., [178])

$$\mathrm{KPI}(\mathbf{P}_0,\boldsymbol{\alpha},\mathbf{H}_\tau) = \sum_{c=1}^{N_C}\sum_{u=1}^{N_U}\log_2\det\left(\mathbf{I}_{N_R} + 10^{\frac{P^{\mathrm{TX}}_{c,u}}{10}}\boldsymbol{\Gamma}^{-1}_{c,u}\mathbf{H}_{\tau,c,u,c}\mathbf{H}^{\mathsf{H}}_{\tau,c,u,c}\right) \quad [\text{bit/s/Hz}], \tag{4.24}$$

where $\mathbf{I}_{N_R}$ is the $N_R \times N_R$ identity matrix, and $\boldsymbol{\Gamma}_{c,u}$ is the noise-plus-interference covariance matrix for the transmission of UE u towards the serving BS in cell c, i.e.,

$$\boldsymbol{\Gamma}_{c,u} = 10^{\frac{\sigma_z^2}{10}}\mathbf{I}_{N_R} + \sum_{j=1,j\neq u}^{N_U} 10^{\frac{P^{\mathrm{TX}}_{c,j}}{10}}\mathbf{H}_{\tau,c,j,c}\mathbf{H}^{\mathsf{H}}_{\tau,c,j,c} + \sum_{c'=1,c'\neq c}^{N_C}\sum_{u=1}^{N_U} 10^{\frac{P^{\mathrm{TX}}_{c',u}}{10}}\mathbf{H}_{\tau,c,u,c'}\mathbf{H}^{\mathsf{H}}_{\tau,c,u,c'}, \tag{4.25}$$

with σ_z^2 as the channel noise power in logarithmic scale. Note that the transmitted powers $P^{\mathrm{TX}}_{c,j}$ from each jth UE in any cell c are also measured in logarithmic scale.

The joint distribution $\mathrm{P}_{\mathbf{H}_\tau}$ of the channel matrices $\mathbf{H}_\tau = \{\mathbf{H}_{\tau,c,u,c'}\}_{c=1,u=1,c'=1}^{N_C,N_U,N_C}$ depends on the wrap-around distance $\{d_{c,u,c'}\}$ between the uth UE in cell c and the BS in cell c' for $u = 1,\ldots,N_U$ and $c,c' = 1,\ldots,N_C$; on the receiver antennas height h_{BS} relative to the UEs' height; on the power of shadow fading σ^2_{SF}; and on the carrier frequency f_c. Specifically, we model the $N_R \times N_T$ channel between UE u in cell c and the BS in cell c'

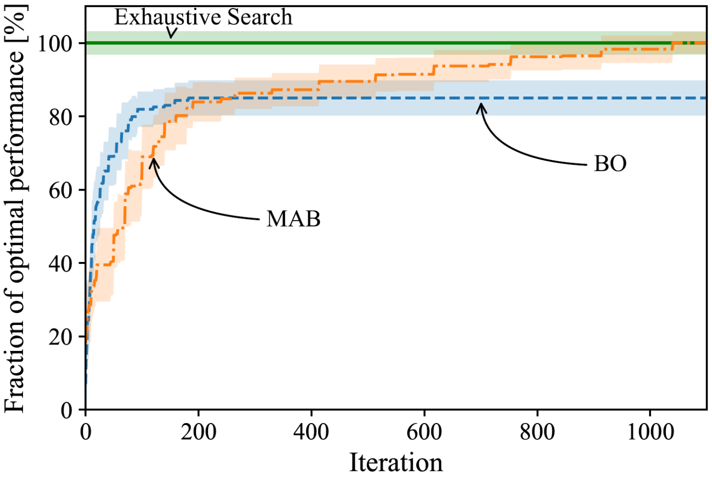

Figure 4.3 Fraction of the optimal KPI (4.24) (compared to exhaustive search) obtained by BO and MAB optimizers for a multi-cell system as a function of the number of iterations of the optimization algorithms.

as

$$\mathbf{H}_{\tau,c,u,c'} = 10^{\frac{-\mathrm{PL}_{\tau,c,u,c'}}{20}} \beta_{\tau,c,u,c'} \mathbf{G}_{\tau,c,u,c'}, \tag{4.26}$$

where the distribution of the $N_R \times N_T$ random matrix $\mathbf{G}_{\tau,c,u,c'}$ and of the coefficient $\beta_{\tau,c,u,c'}$ depend on whether UE u in cell c is in non-line-of-sight (NLOS), or line-of-sight (LOS). With respect to BS c', the LOS probability for each UE u in cell c is computed according to Table 7.4.2-1 in 3GPP TR 38.901 as

$$\mathrm{Pr}_{LOS,\tau,c,u,c'} = \begin{cases} 1 \\ d_{\tau,c,u,c'} \leq d_{min}, \\ \frac{18}{d_{\tau,c,u,c'}} + \exp\left(-\frac{d_{\tau,c,u,c'}}{36}\right)\left(1 - \frac{18}{d_{\tau,c,u,c'}}\right) \\ d_{\tau,c,u,c'} > d_{min}, \end{cases} \tag{4.27}$$

where d_{min} is set to 18 m. The slow fading variable $\beta_{\tau,c,u,c'}$ is log-normal distributed with standard deviations $\sigma_{LOS,\tau}$ or $\sigma_{NLOS,\tau}$ with respective probabilities $\mathrm{Pr}_{LOS,\tau,c,u,c'}$ and $1-\mathrm{Pr}_{LOS,\tau,c,u,c'}$; and the matrix $\mathbf{G}_{\tau,c,u,c'}$ is either Ricean or Rayleigh distributed with respective probabilities $\mathrm{Pr}_{LOS,\tau,c,u,c'}$ and $1-\mathrm{Pr}_{LOS,\tau,c,u,c'}$. Furthermore, the pathloss $\mathrm{PL}_{c,u,c'}$ for LOS and NLOS, which are used in (4.1), are obtained from the urban microcellular (UMi) street canyon pathloss model in Table 7.4.1-1 of 3GPP TR 38.901

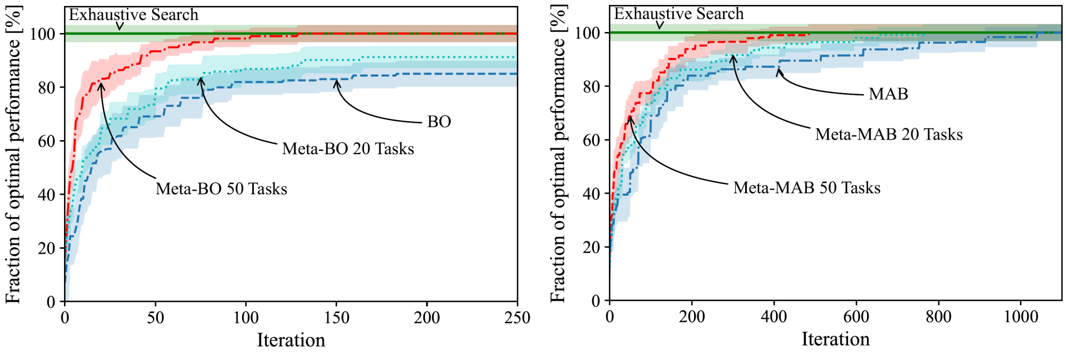

Figure 4.4 Fraction of the optimal KPI (4.24) (compared to exhaustive search) obtained by meta-BO (left) and meta-MAB (right) optimizers for a multi-cell system as a function of the number of iterations of the optimization algorithms.

as

$$\begin{aligned}
&\text{PL}_{LOS,c,u,c'} = 32.4 + 21\log_{10}(d'_{c,u,c'}) + 20\log_{10}(f_c),\\
&\text{PL}_{NLOS,c,u,c'} = \max\Big(\text{PL}_{LOS,c,u,c'}, 35.3\log_{10}(d'_{c,u,c'})\\
&+22.4 + 21.3\log_{10}(f_c) - 0.3(h_{UE} - 1.5)\Big),
\end{aligned} \tag{4.28}$$

respectively, where $d'_{\tau,c,k}$ is the distance between UEs and receiver antennas in the wrap-around model. The parameter $\text{CL}_{u,c}$ in (4.1) is fixed to 0 dB in accordance to Table 7.2.1-1 in 3GPP TS 38.213.

We focus on the optimization of a single pair (P_0, α) of OLPC parameters shared across three cells. This relatively simple setting allows us to maximize function (4.24) exactly through exhaustive search, providing a useful benchmark for the considered approximate optimization strategies.

We fix the number of antennas to $N_R = 16$ and $N_T = 4$, the number of UEs to $N_U = 10$ in each cell, the carrier frequency to $f_c = 3.5$ GHz, the size of the CSI dataset for each configuration τ is set to $S_\tau = 100$ samples, and the maximum transmit power is $P_{MAX,u} = 23$ dBm for all UEs.

For each configuration, the location of the UEs is fixed, and obtained by drawing distances $d_{c,u,c}$ to a serving BS uniformly in the interval $[18, 200]$ meters. As specified in UMi street canyon, the receiver height is $h_{BS} = 15$ meters, the shadow fading standard deviations are set to 4 dB and 7.82 dB. In accordance with Table 7.7.2-4 in 3GPP TR 38.901, Rayleigh fading variance is -13.5 dB for NLOS links, while Rice fading with mean -0.2 dB and variance -13.5 dB affects LOS UEs. The noise power is set to $\sigma_z^2 = -121.38$ dB.

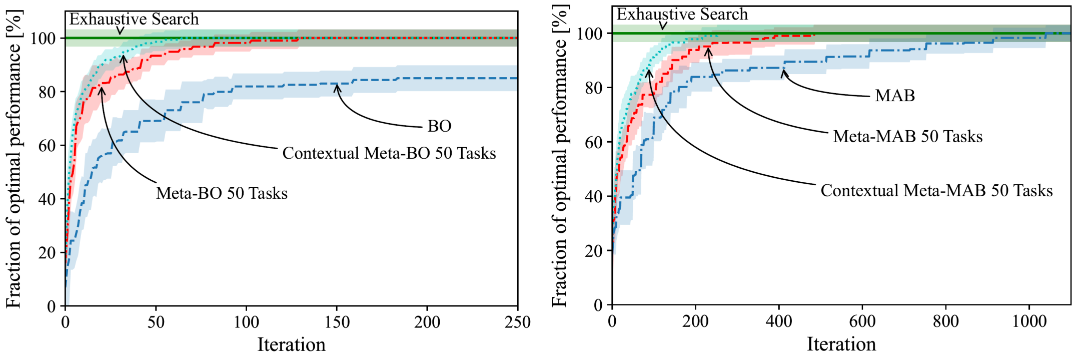

Figure 4.5 Fraction of the optimal KPI (4.24) (compared to exhaustive search) obtained by contextual meta-BO and vanilla meta-BO (left), as wel as by contextual meta-MAB and vanilla meta-MAB (right) for a multi-cell system as a function of the number of iterations of the optimization algorithms.

4.7.2 Conventional Bayesian and Bandit Optimization

First, we evaluate the average KPI function (4.3) using (4.24) in the full (P_0, α) solution space, where the KPI is averaged over 20 realizations of the dataset D_τ with the same configuration τ. Fig. 4.2 shows that the optimization target is multimodal, and hence generally computational challenging for traditional local search algorithms.

We now compare the performance of BO and bandit optimization on a single configuration τ, with the performance averaged over 10 realizations and over 100 CSI datasets for each realization. We plot the KPI value normalized by the optimal value obtained via exhaustive search. The kernels for BO and bandit optimization are selected as *Radial Basis Function kernels* (RBF) with bandwidth tuned to be 0.76 prior to the optimization, and we set parameter $\omega = 0.3$ throughout the experiments for MAB via grid search. BO is seen to outperform bandit optimization for the first several iterations. At later iterations, the performance is limited by the inherent bias of BO due to the continuous model used to approximate optimization in a discrete space. This causes bandit optimization, which operates directly on a discrete space, to outperform BO when the number of iterations is sufficiently large, attaining the performance of exhaustive search.

4.7.3 Bayesian and Bandit Meta-Optimization

Having observed the relative inefficiency of BO and MAB in terms of number of iterations in Fig. 4.3, we now evaluate the performance of Bayesian meta-optimization (Algorithm 2) and bandit meta-optimization (Algorithm 4). We refer to these schemes for short as *meta-BO* and *meta-MAB*, respectively. Both the parametric mean function $\mu_{\boldsymbol{\theta}}(\cdot)$ and function $\psi_{\boldsymbol{\theta}}(\cdot)$ for kernels (4.6) are instantiated as fully-connected neural networks with 3 layers with each 32 neurons. The design guidelines of the configuration of neural network architectures depends on the complexity and quality of service (QoS) requirements of the

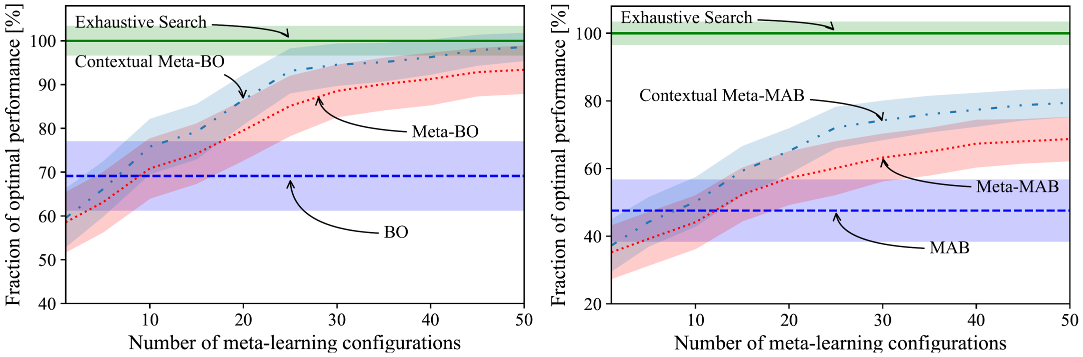

Figure 4.6 Fraction of the optimal KPI (4.24) (compared to exhaustive search) obtained by contextual meta-BO and vanilla meta-BO (left), as wel as by contextual meta-MAB and vanilla meta-MAB (right) for a multi-cell system as a function of the number of available meta-training configurations.

real world scenarios. Setting the number of meta-training configurations to $N = 50$, and the number of collected data pairs to $T_n = 10$, Fig. 4.4 shows the fraction of the optimal KPI for both meta-optimization strategies.

It is observed that meta-learning accelerates the convergence for both BO and MAB. For example, meta-MAB with 50 tasks can achieve a 90% fraction of the optimal performance after around 175 iterations, while conventional MAB would require around 510 iterations. However, as the number of iterations increases, the gain of meta-MAB over MAB vanishes, since MAB is already able to achieve the performance of exhaustive search given its direct optimization in the discrete space.

In this regard, BO stands to gain more from the implementation of meta-learning, since, as seen in Fig. 4.3, the performance of BO is limited by the bias caused by the optimization over a continuous space as the number of iterations increase. For instance, with data from 50 tasks, meta-BO can achieve a 90% fraction of the optimal performance already at 50 iterations, while conventional BO would not be able to obtain this performance level. More generally, meta-BO with 50 tasks can achieve any desired performance level in less than around 150 iterations. This indicates that optimizing the kernel via meta-learning can fully compensate for the bias caused by the fact that BO addresses the optimization problem in a continuous design space.

Overall, while, without meta-learning, MAB is preferable over BO if the goal is achieving high-quality solutions, as long as data from a sufficiently large number of tasks is available, meta-BO becomes significantly advantageous. For the example at hand, as mentioned, a 90% performance level is obtained with meta-MAB with around 175 iterations. while meta-BO requires only 50 iterations.

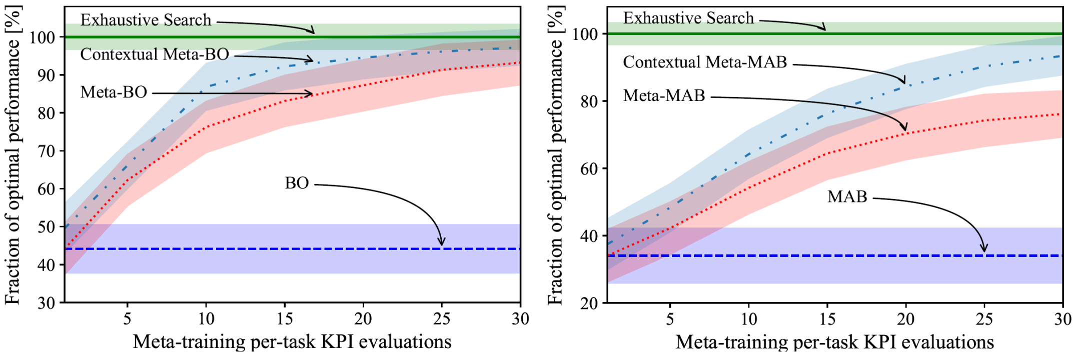

Figure 4.7 Fraction of the optimal KPI (4.24) (compared to exhaustive search) obtained by contextual meta-BO and vanilla meta-BO (left), as wel as by contextual meta-MAB and vanilla meta-MAB (right) for a multi-cell system as a function of the number of available KPI evaluations T_n per-meta-training task.

4.7.4 Contextual Bayesian and Bandit Meta-Optimization

We now investigate the performance of contextual Bayesian meta-optimization (Sec. 4.6.3) and contextual Bandit meta-optimization (Sec. 4.6.4), which we refer for short as *contextual meta-BO* and *contextual meta-MAB*, respectively. We are interested in addressing the potential benefits as compared to vanilla meta-BO and meta-MAB. In order to obtain the interference graph, the threshold ratio d_{ji}/d_i is set to 1.8 (or other values subject to the channel estimations in practice); and the rank of the parameter matrices $\mathbf{V}_1, \mathbf{V}_2$ is set to $r = 14$ for both algorithms. Both values are obtained via a coarse grid search. The number of meta-training tasks is set to $N = 50$.

Fig. 4.5 demonstrates the fraction of optimal KPI for both context-based strategies as compared to the vanilla counterpart solutions. The results validate the capacity of the proposed contextual meta-learning methods to extract useful information from the network topology for the given configuration, achieving faster convergence for both Meta-BO and Meta-MAB.

We elaborate on the impact of the number N of meta-training tasks in Fig. 4.6, which shows the fraction of optimal KPI obtained at the 50th iteration. It is observed that a number of meta-training tasks equal to $N = 10$ for meta-BO and $N = 12$ for meta-MAB is sufficient to ensure that vanilla meta-BO and meta-MAB optimizers can transfer useful information from the meta-training configurations to the new configurations to speed up optimization as compared to BO and MAB, respectively. Furthermore, contextual meta-BO and contextual meta-MAB can further decrease the number of required meta-training configurations.

Finally, we address the impact of the number T_n of per-task KPI evaluations available in the meta-training data. We evaluate the fraction of the optimal KPI obtained at the 20th iteration, and set $N = 50$ tasks. In Fig. 4.7, we observe that meta-BO and meta-MAB, as well as their contextual versions, can significantly enhance the performance of vanilla BO and MAB with as few as $T_n = 20$ KPI evaluations per task. Concretely, while vanilla BO obtains a fraction around 40% of the optimal performance, with $T_n = 20$, contextual BO achieves more than 90% of this fraction, providing a 10% gain over meta-BO. Similarly, while vanilla MAB obtains 30% of the optimal performance, with $T_n = 20$, meta-MAB obtains a 70% fraction, and contextual MAB an 80% fraction.

4.8 Conclusions

Modern cellular networks require complex resource allocation procedures that can only leverage limited access to KPI evaluations for different candidate resource-allocation parameters. While data collection for the current network deployment of interest is challenging, a network operator has typically access to data from related, but distinct, deployments. This chapter has proposed to transfer knowledge from such historical or simulated deployments via an offline meta-learning phased with the aim of learning how to optimize on new deployments. As such, the proposed meta-learning approach can be integrated with digital twin platform providing simulated data [147]. We have specifically focused on BO and MAB optimizers, with the former natively operating on a continuous optimization domain and the latter on a discrete domain. Furthermore, we have proposed novel BO and MAB-based optimizers that can integrate contextual information in the form of interference graphs into the resource-allocation optimization. The extra computational complexity for all proposed methods is of $\mathcal{O}(2NT_{max}^3)$ induced by gradient descent [156] and Gramian matrix inversion in optimizing hyperparameters $\boldsymbol{\theta}^*$ compared to standard BO and MAB. Experimental results have validated the efficiency gains of meta-learning and contextual meta-learning.

Future work may address online meta-learning techniques that successively improve the efficiency of resource allocation as data from more deployments is (see [135] for a related application to demodulation and [115] to drone trajectory optimization). Moreover, it would be interesting to investigate the application to larger-scale problems involving real-world data; the extension to multi-objective problems [175]; and the interplay with digital twin platforms for the management of wireless systems [147]. From the perspective of more practical applications in the next generation wireless communications, the communication models considered in this chapter can also be extended to more complicated dynamic mobile edge computing (MEC) where joint optimization of discrete offloading tasks and analog resource allocation are required [195], or resource allocation for uplink rate splitting multiple access (RSMA) in future 6G wireless networks [70].

Chapter 5

Multi-Fidelity Bayesian Optimization With Across-Task Transferable Max-Value Entropy Search

5.1 Overview

In this Chapter, we consider the scenarios where optimization tasks are modelled in a multi-fidelity manner and arrive in a sequence at the optimizer. In many applications, ranging from logistics to engineering, a designer is faced with a sequence of optimization tasks for which the objectives are in the form of black-box functions that are costly to evaluate. For example, the designer may need to tune the hyperparameters of neural network models for different learning tasks over time. Rather than evaluating the objective function for each candidate solution, the designer may have access to approximations of the objective functions, for which higher-fidelity evaluations entail a larger cost. Existing multi-fidelity black-box optimization strategies select candidate solutions and fidelity levels with the goal of maximizing the information accrued about the optimal value or solution for the current task. Assuming that successive optimization tasks are related, this chapter introduces a novel information-theoretic acquisition function that balances the need to acquire information about the current task with the goal of collecting information transferable to future tasks. The proposed method includes shared inter-task latent variables, which are transferred across tasks by implementing particle-based variational Bayesian updates. Experimental results across synthetic and real-world examples reveal that the proposed provident acquisition strategy that caters to future tasks can significantly improve the optimization efficiency as soon as a sufficient number of tasks is processed.

5.2 Introduction

5.2.1 Context and Scope

Numerous problems in logistics, science, and engineering can be formulated as *black-box optimization* tasks, in which the objective is costly to evaluate. Examples include hyperparameter optimization for machine learning [203], malware detection [32], antenna design [210], text to speech adaptation [124], material discovery [194], and resource allocation in wireless communication systems [112, 208, 185, 73]. To mitigate the problem of evaluating a costly objective function for each candidate solutions, the designer may have access to cheaper *approximations* of the optimization target. For example, the designer may be able to *simulate* a physical system using a *digital twin* that offers a controllable trade-off between *cost* and *fidelity* of the approximation [63, 68, 147, 148]. As shown in Fig. 5.1, higher-fidelity evaluations of the objective functions generally entail a larger cost, and the main challenge for the designer is to select a sequence of candidate solutions and fidelity levels that obtains the best solution within the available cost budget.

As a concrete example, consider the problem of optimizing the time spent by patients in a hospital's emergency department [23]. The hospital may try different allocations of medical personnel by carrying out expensive real-world trials. Alternatively, one may adopt a simulator of patients' hospitalization experiences, with different accuracy levels requiring a larger computing cost in terms of time and energy.

As also illustrated in Fig. 5.1, in many applications, the designer is faced with a *sequence* of black-box optimization tasks for which the objectives are distinct, but related. For instance, one may need to tune the hyperparameters of neural network models for different learning tasks over time; address the optimal allocation of personnel in a hospital in different periods of the year; or optimize resource allocation in a wireless system as the users' demands change over time. As detailed in the next section, existing multi-fidelity black-box optimization strategies select candidate solutions and fidelity levels with the goal of maximizing the information accrued about the optimal value or solution for the *current* task.

This chapter introduces a novel information-theoretic selection process for the next candidate solution and fidelity level that balances the need to acquire information about the *current* task with the goal of collecting information transferable to *future* tasks. The proposed method introduces *shared* latent variables across tasks. These variables are transferred across successive tasks by adopting a Bayesian formalism whereby the posterior distribution at the end of the current task is adopted as prior for the next task.

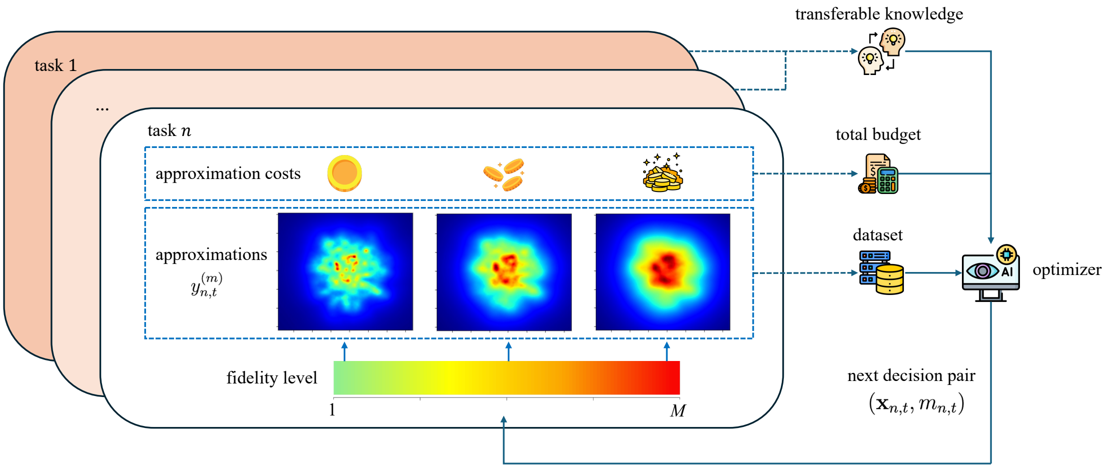

Figure 5.1 This chapter studies a sequential multi-task optimization setting with multi-fidelity approximations of expensive-to-evaluate black-box objective functions. For any current task n, over time index $t = 1, 2, ..., T_n$, the optimizer selects a pair of query point $\mathbf{x}_{n,t}$ and fidelity level $m_{n,t}$, requiring an approximation cost $\lambda^{(m)}$. As a result, the optimizer receives noisy feedback $y_{n,t}^{(m)}$ about target objective value $f_n(\mathbf{x}_{n,t})$. We wish to approach the global optimal solution $\mathbf{x}_n^*$ of objective $f_n(\mathbf{x})$ while abiding by a total simulation cost budget Λ.

5.2.2 Related Work

BO is a popular framework for black-box optimization problems. BO relies on a *surrogate model*, typically a GP [142], which encodes the current belief of the optimizer about the objective function, and an *acquisition function* that selects the next candidate solution based on the surrogate model [74, 48, 189, 188, 162, 207]. BO has been extended to address multi-fidelity – also known as multi-task or multi-information source – settings [125, 172, 141]. Via *multi-fidelity BO* (MFBO), information collected at lower fidelity levels can be useful to accelerate the optimization process when viewed as a function of the overall cost budget for evaluating the objective function.

As illustrated in Fig. 5.2, the other axis of generalization of BO of interest for this work, namely *transferability* across tasks, has been much less investigated. This line of work relies on the assumption that sequentially arriving optimization tasks are statistically correlated, such that knowledge extracted from one task can be transferred to future tasks in the form of an optimized inductive bias encoded into the optimizer [181]. Existing studies fall into the categories of *lifelong BO*, which leverages previously trained deep neural networks to accelerate the optimizer training process exclusively on the new task [206]; and *meta-learned BO*, which learns a well-calibrated prior on the surrogate model given datasets collected from previous tasks [208, 146]. Using these methods, BO is seen to successfully transfer shared information across tasks, providing faster convergence on later tasks. However, these studies are limited to single-fidelity settings.

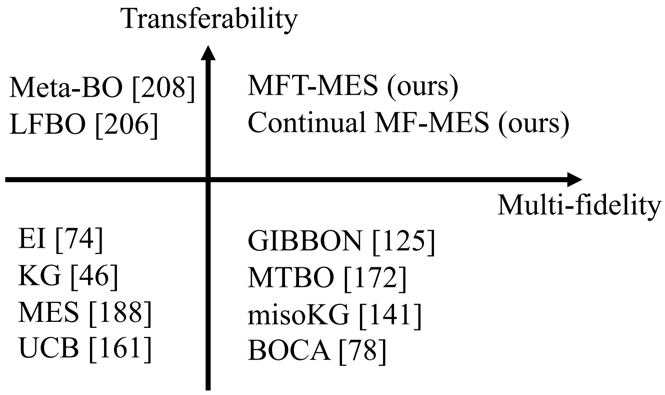

Figure 5.2 Comparison between the state-of-the-art methods and the proposed MFT-MES approach.

Another related line of work corresponds to the Bayesian active learning assisted BO, where candidate solutions selected within the target optimization task are considered to contribute to accurately learn the black-box function globally. In the information-theoretic view, *active learning McKay* (ALM) [109] searches for areas with maximum Shannon entropy, which for Bayesian surrogate models amounts to inputs with highest uncertainty. *Bayesian active learning by disagreement* (BALD) [67] is the first active learning assisted BO framework that explicitly reduces the uncertainty induced by model hyperparameters. While *Bayesian query-by-committee* (BQBC) [143] selects points where the variance of the mean estimate of surrogate model is maximized with respect to the exchanges of model hyperparameters. However, none of the above existing works considers scenarios where optimization tasks arrives in a stream with limited evaluation budget.

5.2.3 Main Contributions

Assuming that successive optimization tasks are related, this chapter introduces a novel *information-theoretic* acquisition function that balances the need to acquire information about the current task with the goal of collecting information transferable to future tasks. The proposed method, referred to as *multi-fidelity transferable max-value entropy search* (MFT-MES), includes shared inter-task latent variables, which are transferred across tasks by implementing particle-based variational Bayesian updates.

The main contributions are as follows.

- We introduce the MFT-MES, a novel black-box optimization scheme tailored for settings in which the designer is faced with a sequence of related optimization tasks. MFT-MES builds on MF-MES [125] by selecting candidate solutions and fidelity levels that maximize the information gain per unit cost. The information gain in MFT-MES accounts not only for the information about the optimal value of the current objective, as in MF-MES, but also for the information accrued on the inter-task shared latent parameters that can be transferred to future tasks. To this end, MFT-MES models the latent parameters as random quantities whose distributions are updated and transferred across tasks.

- As an efficient implementation of MFT-MES, we propose a particle-based *variational inference* (VI) update strategy for the latent shared parameters by leveraging Stein variational gradient descent (SVGD) [104].
- We present experimental results across synthetic tasks [34] and real-world examples [208, 98]. The results reveal that the provident acquisition strategy implemented by MFT-MES, which caters to future tasks, can significantly improve the optimization efficiency as soon as a sufficient number of tasks is processed.

The rest of this chapter is organized as follows. Sec. 5.3 formulates the sequential multi-task black-box optimization problem, and reviews the MF-GP surrogate model considered in the paper. Sec. 5.4 presents the baseline implementation of MF-MES, and illustrates the optimization over surrogate model parameters. The proposed MFT-MES method and the Bayesian update of the shared parameters are introduced in Sec. 5.5. Experimental results on synthetic optimization tasks and real-world applications are provided in Sec. 5.6. Finally, Sec. 5.7 concludes this chapter.

5.3 Problem Definition And Preliminaries

5.3.1 Sequential Multi-Task Black-Box Optimization

We consider a setting in which optimization tasks, defined on a common input space $\mathcal{X} \subseteq \mathbb{R}^d$, are addressed sequentially. Each n-th task, with $n = 1, 2, ...$, consists of the optimization of a *black-box expensive-to-evaluate* objective function $f_n(\mathrm{x})$. Examples include the optimization of hyperparameters for machine learning models and experimental design [96, 113]. The objective functions $f_n(\mathrm{x})$ are assumed to be drawn according to a common parametric stochastic process $\mathcal{P}_{\boldsymbol{\theta}}(f(\mathrm{x}))$ in an independent identical distributed (i.i.d.) manner, i.e.,

$$f_n(\mathrm{x}) \underset{\text{i.i.d.}}{\sim} \mathcal{P}_{\boldsymbol{\theta}}(f(\mathrm{x})) \text{ for } n = 1, 2, ... \tag{5.1}$$

Furthermore, the parameter vector $\boldsymbol{\theta}$ identifying the stochastic process $\mathcal{P}_{\boldsymbol{\theta}}(f(\mathrm{x}))$ is unknown, and it is assigned a prior distribution $p(\boldsymbol{\theta})$, i.e.,

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}). \tag{5.2}$$

Note that, by (5.1) and (5.2), the objective function $f_1(\mathbf{x}), f_2(\mathbf{x}), ...$ are not independent, since having information on any function $f_n(\mathbf{x})$ would reduce uncertainty on the parameters $\boldsymbol{\theta}$, thus also providing information about other function $f_{n'}(\mathbf{x})$ with $n' \neq n$.

For any current n-th task, the goal is to obtain an approximation of the optimal solution

$$\mathbf{x}_n^* = \arg\max_{\mathbf{x} \in \mathcal{X}} f_n(\mathbf{x}) \tag{5.3}$$

with the minimal number of evaluations of function $f_n(\mathbf{x})$. To this end, this chapter investigates the idea of selecting candidate solution $\mathbf{x}$ to query the current function $f_n(\mathbf{x})$ not only with the aim of approaching the solution in (5.3) for task n, but also to extract information about the common parameters $\boldsymbol{\theta}$ that may be useful for future optimization tasks $n' > n$. This way, while convergence to a solution (5.3) may be slower for the current task n, future tasks may benefit from the acquired knowledge about parameters $\boldsymbol{\theta}$ to speed up convergence.

In order to account for the *cost* of accessing the objective function $f_n(\mathbf{x})$, we follow the *multi-fidelity* formulation, whereby evaluating function $f_n(\mathbf{x})$ at some querying point $\mathbf{x}$ with fidelity level m entails a cost $\lambda^{(m)} > 0$ [44, 92]. Different fidelity levels may correspond to training processes with varying number of iterations for hyperparameters optimization, or to simulations of a physical process with varying levels of accuracy for experimental design.

There are M fidelity levels, listed from lower fidelity, $m = 1$, to highest fidelity, $m = M$, which are collected in set $\mathcal{M} = \{1, 2, ..., M\}$. The function approximating objective $f_n(\mathbf{x})$ at the m-th fidelity level is denoted as $f_n^{(m)}(\mathbf{x})$. The costs are ordered from lowest fidelity to highest fidelity as

$$\lambda^{(1)} \leq \lambda^{(2)} \leq ... \leq \lambda^{(M)}, \tag{5.4}$$

and the highest-fidelity approximation coincides with the true objective function, i.e.,

$$f_n^{(M)}(\mathbf{x}) = f_n(\mathbf{x}). \tag{5.5}$$

For each task n, the optimizer queries the objective function $f_n(\mathbf{x})$ during T_n *rounds*, choosing at each round $t = 1, ..., T_n$, an input $\mathbf{x}_{n,t}$ and a fidelity level $m_{n,t}$. The number of rounds, T_n, is dictated by a cost budget, to be introduced below. The corresponding observation is given as

$$y_{n,t}^{(m)} = f_n^{(m)}(\mathbf{x}_{n,t}) + \epsilon_{n,t}, \tag{5.6}$$

where the observation noise variables $\epsilon_{n,t} \sim \mathcal{N}(0, \sigma^2)$ are independent. Each pair $(\mathbf{x}_{n,t}, m_{n,t})$ is chosen by the optimizer based on the past observations

$$\mathcal{D}_{n,t} = \left\{ (\mathbf{x}_{n,1}, m_{n,1}, y_{n,1}^{(m_1)}), ..., (\mathbf{x}_{n,t}, m_{n,t}, y_{n,t}^{(m_t)}) \right\} \tag{5.7}$$

for the current task n, as well as based on the dataset

$$\mathcal{D}_{n-1} = \bigcup_{n'=1}^{n-1} \mathcal{D}_{n',T_{n'}} \tag{5.8}$$

collected for all the previous tasks $n' = 1, ..., n-1$. In practice, as we will see, data set $\mathcal{D}_{n-1}$ need not be explicitly stored. Rather, information in $\mathcal{D}_{n-1}$ that is useful for future tasks is summarized into a distribution over the shared parameter vector $\boldsymbol{\theta}$.

The number of rounds T_n is determined by the *cost constraint*

$$\sum_{t=1}^{T_n} \lambda^{(m_t)} \leq \Lambda \tag{5.9}$$

for each task n, where Λ is a pre-determined total query cost budget for each optimization task. Accordingly, the number of rounds T_n is the maximum integer such that constraint (5.9) is satisfied.

5.3.2 Gaussian Process

The proposed approach builds on multi-fidelity GPs. To explain, we begin in this subsection with a brief review of *conventional GP*, which corresponds to the special case $M = 1$ of a *single fidelity level*. With $M = 1$, the optimizer maintains a single surrogate objective function for the true objective $f_n^{(1)}(\mathbf{x}) = f_n(\mathbf{x})$ of each task n. In BO, this is done by assigning a zero-mean GP distribution $p(f_n|\boldsymbol{\theta})$ to function $f_n(\mathbf{x})$ that is characterized by a *kernel function* $k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')$, as denoted by

$$f_n \sim \mathcal{GP}(0, k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')). \tag{5.10}$$

The notation $k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')$ makes it clear that the kernel function, measuring the correlation of function values $f_n(\mathbf{x})$ and $f_n(\mathbf{x}')$, depends on the common parameters $\boldsymbol{\theta}$ in (5.2).

By definition of GP, the collection of objective values $\mathbf{f}_{n,t} = [f_n(\mathbf{x}_{n,1}), ..., f_n(\mathbf{x}_{n,t})]$ from any set of inputs $\mathbf{X}_{n,t} = [\mathbf{x}_{n,1}, ..., \mathbf{x}_{n,t}]$ follows a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t}))$, with $t \times 1$ zero mean vector $\mathbf{0}$, and $t \times t$ covariance matrix $\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t})$

given by

$$\mathbf{K}_{\boldsymbol{\theta}_n}(\mathbf{X}_{n,t}) = \begin{bmatrix} k_{\boldsymbol{\theta}_n}(\mathbf{x}_{n,1},\mathbf{x}_{n,1}) & \dots & k_{\boldsymbol{\theta}_n}(\mathbf{x}_{n,1},\mathbf{x}_{n,t}) \\ \vdots & \ddots & \vdots \\ k_{\boldsymbol{\theta}_n}(\mathbf{x}_{n,t},\mathbf{x}_{n,1}) & \dots & k_{\boldsymbol{\theta}_n}(\mathbf{x}_{n,t},\mathbf{x}_{n,t}) \end{bmatrix}. \tag{5.11}$$

A typical parametric kernel function is given by [144]

$$k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') = \exp\left(-||\psi_{\boldsymbol{\theta}}(\mathbf{x}) - \psi_{\boldsymbol{\theta}}(\mathbf{x}')||_2^2\right), \tag{5.12}$$

where $\psi_{\boldsymbol{\theta}}(\cdot)$ is a neural network with parameters $\boldsymbol{\theta}$.

Given any observation history $\mathcal{D}_{n,t}$ for task n in (5.7), and given a parameter vector $\boldsymbol{\theta}$, the posterior distribution of objective value $f_n(\mathbf{x})$ at any input $\mathbf{x}$ is the Gaussian distribution [142]

$$p_{\boldsymbol{\theta}}(f_n(\mathbf{x})|\mathcal{D}_{n,t}) = \mathcal{N}(\mu_{\boldsymbol{\theta}}(\mathbf{x}|\mathcal{D}_{n,t}), \sigma^2_{\boldsymbol{\theta}}(\mathbf{x}|\mathcal{D}_{n,t})), \tag{5.13}$$

where

$$\mu_{\boldsymbol{\theta}}(\mathbf{x}|\mathcal{D}_{n,t}) = \mathbf{k}_{\boldsymbol{\theta}}(\mathbf{x})^{\mathsf{T}}(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t}))^{-1}\mathbf{y}_{n,t}, \tag{5.14}$$

$$\text{and} \quad \sigma^2_{\boldsymbol{\theta}}(\mathbf{x}|\mathcal{D}_{n,t}) = k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}) - \mathbf{k}_{\boldsymbol{\theta}}(\mathbf{x})^{\mathsf{T}}(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t}))^{-1}\mathbf{k}_{\boldsymbol{\theta}}(\mathbf{x}), \tag{5.15}$$

with the $t \times t$ Gramian matrix $\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t}) = \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t}) + \sigma^2\mathbf{I}_t$; the $t \times 1$ cross-variance vector $\mathbf{k}_{\boldsymbol{\theta}}(\mathbf{x}) = [k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}_{n,1}),\dots,k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}_{n,t})]^{\mathsf{T}}$; and the $t \times 1$ observations vector $\mathbf{y}_{n,t} = [y_{n,1},...,y_{n,t}]^{\mathsf{T}}$.

5.3.3 Multi-fidelity Gaussian Process

Multi-fidelity Gaussian Process (MFGP) provides a surrogate model for the objective functions $(f_n^{(1)}(\mathbf{x}),...,f_n^{(M)}(\mathbf{x}))$ across all M fidelity levels [15, 52, 82]. This is done by defining a kernel function of the form $k_{\boldsymbol{\theta}}\big((\mathbf{x},m),(\mathbf{x}',m')\big)$ that captures the correlations between the function values $f_n^{(m)}(\mathbf{x})$ and $f_n^{(m')}(\mathbf{x}')$ for any two inputs $\mathbf{x}$ and $\mathbf{x}'$, and for any two fidelity levels m and m'. Examples of such kernels include the co-kriging model in [82] and the *intrinsic coregionalization model* (ICM) kernel [15, 2], which is expressed as

$$k_{\boldsymbol{\theta}}\big((\mathbf{x},m),(\mathbf{x}',m')\big) = k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') \cdot \kappa_{\boldsymbol{\theta}}(m,m'), \tag{5.16}$$

where the input-space kernel $k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}')$ is defined as in the previous subsection; while the fidelity space kernel $\kappa_{\boldsymbol{\theta}}(m,m')$ is often instantiated as a RBF kernel

$$\kappa(m,m') = \exp(-\gamma||m-m'||^2), \tag{5.17}$$

where the bandwidth parameter γ is included in the hyperparameters $\boldsymbol{\theta}$. The MFGP prior for functions $f_n^{(1)}(\mathbf{x}),...,f_n^{(M)}(\mathbf{x})$ is denoted as

$$\left(f_n^{(1)}(\mathbf{x}),...,f_n^{(M)}(\mathbf{x})\right) \sim \mathcal{GP}(0, k_{\boldsymbol{\theta}}((\mathbf{x},m),(\mathbf{x}',m'))). \tag{5.18}$$

Let $\mathbf{y}_{n,t} = [y_{n,1}^{(m_1)},...,y_{n,t}^{(m_t)}]^\mathsf{T}$ denote the $t \times 1$ observations column vector, and $\mathbf{m}_{n,t} = [m_{n,1},...,m_{n,t}]^\mathsf{T}$ be the $t \times 1$ vector of queried fidelity levels. Using the $t \times t$ kernel matrix $\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t},\mathbf{m}_{n,t})$ in which the (i,j)-th element $k_{\boldsymbol{\theta}}((\mathbf{x}_{n,i},m_{n,i}),(\mathbf{x}_{n,j},m_{n,j}))$ is defined as in (5.16), the MFGP posterior mean and variance at any input $\mathbf{x}$ with fidelity m given the observation history $\mathcal{D}_{n,t}$ can be expressed as [174]

$$\mu_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}|\mathcal{D}_{n,t}) = \mathbf{k}_{\boldsymbol{\theta}}(\mathbf{x},m)^\mathsf{T}(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t},\mathbf{m}_{n,t}))^{-1}\mathbf{y}_{n,t}, \tag{5.19}$$

$$[\sigma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}|\mathcal{D}_{n,t})]^2 = k_{\boldsymbol{\theta}}((\mathbf{x},m),(\mathbf{x},m)) - \mathbf{k}_{\boldsymbol{\theta}}(\mathbf{x},m)^\mathsf{T}(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t},\mathbf{m}_{n,t}))^{-1}\mathbf{k}_{\boldsymbol{\theta}}(\mathbf{x},m), \tag{5.20}$$

where $\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t},\mathbf{m}_{n,t}) = \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t},\mathbf{m}_{n,t}) + \sigma^2\mathbf{I}_t$ is the $t \times t$ Gramian matrix.

Accordingly, an estimated value of the objective $f_n^{(m)}(\mathbf{x})$ can be obtained as the mean $\mu_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}|\mathcal{D}_{n,t})$ in (5.19), and the corresponding uncertainty of the estimate can be quantified by the variance $[\sigma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}|\mathcal{D}_{n,t})]^2$ in (5.20).

5.4 Single-Task Multi-fidelity Bayesian Optimization

In this section, we review a baseline implementation of MFBO based on MES [188, 125], which applies separately to each task n, without attempting to transfer knowledge across tasks.

5.4.1 Multi-Fidelity Max-Value Entropy Search

For brevity of notation, we henceforth omit the dependence on the observation history $\mathcal{D}_{n,t}$ of the MFGP posterior mean $\mu_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}|\mathcal{D}_{n,t})$ and variance $[\sigma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}|\mathcal{D}_{n,t})]^2$ in (5.19) and (5.20), writing $\mu_{\boldsymbol{\theta}}^{(m)}(\mathbf{x})$ and $[\sigma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x})]^2$, respectively. Throughout this subsection, the parameter vector $\boldsymbol{\theta}$ is fixed, and the selection of $\boldsymbol{\theta}$ is discussed in the next subsection. In general *multi-fidelity max-value entropy search* (MF-MES) [125], at each time t, the next input $\mathbf{x}_{n,t+1}$ is selected, together with the fidelity level $m_{n,t+1}$, so as to maximize the ratio between the *informativeness* of the resulting observation $y_n^{(m)}$ in (5.6) and the cost $\lambda^{(m)}$.

Informativeness is measured by the *mutual information* between the optimal value $f_n(\mathbf{x}_n^*) = f_n^*$ of the objective and the observation $y_n^{(m)}$ corresponding to input $\mathbf{x}_{n,t+1}$ at fidelity level m. Accordingly, the next pair $(\mathbf{x}_{n,t+1}, m_{n,t+1})$ is obtained by maximizing the information gain per cost unit as

$$(\mathbf{x}_{n,t+1}, m_{n,t+1}) = \arg\max_{\substack{\mathbf{x}\in\mathcal{X}\\ m\in\mathcal{M}}} \frac{I(f_n^*; y_n^{(m)}|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t})}{[\lambda^{(m)}]^{\omega}}, \tag{5.21}$$

where the scaling power ω depends on the simulation cost model design. In (5.21), the mutual information is evaluated with respect to the joint distribution

$$p(f_n^*, f_n^{(m)}(\mathbf{x}), y_n^{(m)}|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t}) = p(f_n^*, f_n^{(m)}(\mathbf{x})|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t}) p(y_n^{(m)}|f_n^{(m)}(\mathbf{x})), \tag{5.22}$$

where $p(f_n^*, f_n^{(m)}(\mathbf{x})|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t})$ follows the posterior GP expressed by (5.19) as well as (5.20), and $p(y_n^{(m)}|f_n^{(m)}(\mathbf{x}))$ is defined by the observation model (5.6). Note that, to evaluate (5.21), the true, unobserved, function value $f_n^{(m)}(\mathbf{x})$ must be marginalized over.

Let us write as $H(\mathbf{y}|\mathbf{x})$ for the differential entropy of a variable $\mathbf{y}$ given a variable $\mathbf{x}$. By definition, the mutual information in (5.21) can be expressed as the difference of differential entropies [27]

$$\begin{aligned} &I(f_n^*; y_n^{(m)}|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t}) \\ &= H(y_n^{(m)}|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t}) - \mathbb{E}_{p(f_n^*|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t})}[H(y_n^{(m)}|f_n^*, \mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t})] \\ &= \log(\sqrt{2\pi e}\sigma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x})) - \mathbb{E}_{p(f_n^*|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t})}[H(y_n^{(m)}|f_n^{(m)}(\mathbf{x}) \leq f_n^*, \mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t})], \end{aligned} \tag{5.23}$$

where the variance $[\sigma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x})]^2$ is as in (5.20), and (5.23) relies on the assumption that the m-th surrogate function $f_n^{(m)}(\mathbf{x})$ cannot attain a value larger than the maximum f_n^* of the true objective function $f_n(\mathbf{x})$. Alternatively, one can remove this assumption by adopting a more complex approximation illustrated as in [174]. In MF-MES, the second term in (5.23) is approximated as [125]

$$\begin{aligned} &H(y_n^{(m)}|f_n^{(m)}(\mathbf{x}) \leq f_n^*, \mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t}) \\ &\approx \log\left(\sqrt{2\pi e}\sigma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x})\left(1 - \frac{\phi(\gamma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}, f_n^*))}{\Phi(\gamma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}, f_n^*))}\left[\gamma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}, f_n^*) + \frac{\phi(\gamma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}, f_n^*))}{\Phi(\gamma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}, f_n^*))}\right]\right)\right) \\ &= H_{\boldsymbol{\theta}}^{\text{MF-MES}}(\mathbf{x}, m, f_n^*) \end{aligned} \tag{5.24}$$

$$\text{with} \quad \gamma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}, f_n^*) = \frac{f_n^* - \mu_{\boldsymbol{\theta}}^{(m)}(\mathbf{x})}{\sigma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x})}, \tag{5.25}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the probability density function and cumulative density function of a standard Gaussian distribution, respectively. Intuitively, function $H_{\boldsymbol{\theta}}^{\text{MF-MES}}(\mathbf{x}, m, f_n^*)$ in (5.25) captures the uncertainty on the observation $y_n^{(m)}$ that would be produced by querying the objective at input $\mathbf{x}$ and fidelity level m when the optimal value f_n^* is known. This uncertainty should be subtracted, as per (5.23), from the overall uncertainty

Algorithm 5: Multi-Fidelity Max-Value Entropy Search (MF-MES) [125]

```
Input : Vector $\boldsymbol{\theta}$, simulation costs $\{\lambda^{(m)}\}_{m=1}^{M}$, query budget $\Lambda$
Output : Optimized solution $\mathbf{x}^{\text{opt}}$
Initialize iteration $t = 0$, observation dataset $\mathcal{D}_{n,t} = \emptyset$, and $\Lambda^0 = \Lambda$
while $\Lambda^t > 0$ do
    Sample max-value set $\mathcal{F}$ from distribution $p_{\boldsymbol{\theta}}(f_n^*|\mathbf{x}, \mathcal{D}_{n,t})$
    Obtain the next decision pair $(\mathbf{x}_{n,t+1}, m_{n,t+1})$ via (5.26)
    Observe $y_{n,t+1}^{(m)}$ in (5.6) and update observation history
      $\mathcal{D}_{n,t+1} = \mathcal{D}_{n,t} \cup (\mathbf{x}_{n,t+1}, m_{n,t+1}, y_{n,t+1}^{(m_{n,t+1})})$
    Update the MFGP posterior as defined by (5.19) and (5.20)
    Calculate remaining budget $\Lambda^{(t+1)} = \Lambda^{(t)} - \lambda^{(m_{n,t+1})}$
    Set iteration $T_n = t$ and $t = t + 1$
end
Return $\mathbf{x}^{\text{opt}} = \arg\max_{t=1,..,T_n} f_n^{(M)}(\mathbf{x}_{n,t})$
```

$H(y_n^{(m)}|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t})$ in order to assess the extent to which the observation $y_n^{(m)}$ provides information about the optimal value f_n^*.

Using (5.24) in (5.23) and replacing the expectation in (5.23) with an empirical average, MF-MES selects the next query as

$$(\mathbf{x}_{n,t+1}, m_{n,t+1}) = \arg\max_{\substack{\mathbf{x}\in\mathcal{X} \\ m\in\mathcal{M}}} \alpha_{\boldsymbol{\theta}}^{\text{MF-MES}}(\mathbf{x}, m), \tag{5.26}$$

where we have defined the MF-MES acquisition function

$$\alpha_{\boldsymbol{\theta}}^{\text{MF-MES}}(\mathbf{x}, m) = \frac{1}{[\lambda^{(m)}]^{\omega}} \log(\sqrt{2\pi e}\sigma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x})) - \frac{1}{|\mathcal{F}|[\lambda^{(m)}]^{\omega}} \sum_{f_n^* \in \mathcal{F}} H_{\boldsymbol{\theta}}^{\text{MF-MES}}(\mathbf{x}, m, f_n^*), \tag{5.27}$$

where the set $\mathcal{F} = \{f_{n,s}^*\}_{s=1}^{S}$ collects S samples drawn from distribution $p_{\boldsymbol{\theta}}(f_n^*|\mathbf{x}, \mathcal{D}_{n,t})$, which can be obtained via Gumbel sampling [188] and function $H_{\boldsymbol{\theta}}^{\text{MF-MES}}(\mathbf{x}, m, f_n^*)$ is defined in (5.24).

The overall procedure of MF-MES is summarized in Algorithm 5.

5.4.2 Optimizing the Kernel Parameter Vector

In Sec. 5.4.1, we have treated the parameter vector $\boldsymbol{\theta}$ as fixed. In practice, given the data set $\mathcal{D}_{n,t}$ collected up to round t for the current task n, it is possible to update the parameter vector $\boldsymbol{\theta}$ to fit the available observations [142]. This is typically done by maximizing the marginal likelihood of parameters $\boldsymbol{\theta}$ given data $\mathcal{D}_{n,t}$.

Under the posterior distribution defined by (5.19) and (5.20), the *negative marginal log-likelihood* of the parameter vector $\boldsymbol{\theta}$ is given by

$$\begin{aligned}\ell(\boldsymbol{\theta}|\mathcal{D}_{n,t}) &= -\log\big(p(\mathbf{y}_{n,t}|\boldsymbol{\theta})\big)\\ &= -\frac{1}{2}\left(t\log 2\pi - \log\left|\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t},\mathbf{M}_{n,t})\right| - \mathbf{y}_{n,t}^{\mathsf{T}}\Big(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}_{n,t},\mathbf{M}_{n,t})\Big)^{-1}\mathbf{y}_{n,t}\right).\end{aligned} \tag{5.28}$$

where $p(\mathbf{y}_{n,t}|\boldsymbol{\theta})$ represents the probability density function of observation in (5.6). The negative marginal log-likelihood (5.28) can be interpreted as a loss function associated with parameters $\boldsymbol{\theta}$ based on the observations in data set $\mathcal{D}_{n,t}$. Accordingly, using the prior distribution (5.2), a *maximum a posterior* (MAP) solution for the parameter vector $\boldsymbol{\theta}$ is obtained by addressing the problem

$$\boldsymbol{\theta}_{n,t}^{\mathrm{MAP}} = \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\{\ell(\boldsymbol{\theta}|\mathcal{D}_{n,t}) - \log\big(p(\boldsymbol{\theta})\big)\}, \tag{5.29}$$

where $p(\boldsymbol{\theta})$ is the prior distribution in (5.2), and the term $\log\big(p(\boldsymbol{\theta})\big)$ plays the role of a regularizer. To reduce computational complexity, the optimization problem (5.29) may be addressed periodically with respect to the round index t [31, 112].

5.5 Sequential Multi-fidelity BO With Transferable Max-Value Entropy Search

As reviewed in the previous section, MF-MES treats each task n separately [174, 125]. However, since by (5.1) and (5.2), the successive objective functions $f_1,...,f_n$ are generally correlated, knowledge extracted from one task can be transferred to future tasks through the common parameters $\boldsymbol{\theta}$ [159]. In this section, we introduce a novel information-theoretic acquisition function that generalizes the MF-MES acquisition function in (5.21) to account for the information acquired about parameters $\boldsymbol{\theta}$ for future tasks. The proposed approach, referred to as *multi-fidelity transferable max-value entropy search* (MFT-MES), hinges on a Bayesian formulation for the problem of sequentially estimating parameter vector $\boldsymbol{\theta}$ as more tasks are observed. In this section, we first describe the proposed acquisition function, and then describe an efficient implementation of the Bayesian estimation of parameter vector $\boldsymbol{\theta}$ based on *Stein variational gradient descent* (SVGD) [104].

5.5.1 Multi-Fidelity Transferable Max-Value Entropy Search

MFT-MES introduces a term in the MF-MES acquisition function (5.21) that promotes the selection of inputs $\mathbf{x}$ and fidelity levels m that maximize the information brought by the corresponding observation $y_{n,t}^{(m)}$ about the parameters $\boldsymbol{\theta}$. The rationale behind this modification is that collecting information about the shared parameters $\boldsymbol{\theta}$ can potentially improve the optimization process for future tasks $n' > n$.

Accordingly, MFT-MES adopts an acquisition function that measures the mutual information between the observation $y_n^{(m)}$ and, not only the optimal value f_n^* for the current task n in MF-MES, but also the common parameters $\boldsymbol{\theta}$. Note that, unlike MF-MES, this approach views the parameter vector $\boldsymbol{\theta}$ as a random quantity that is jointly distributed with the objective and with the observations. Normalizing by the cost $\lambda^{(m)}$ as in (5.21), MFT-MES selects the next query $(\mathbf{x}_{n,t+1}, m_{n,t+1})$ with the aim of addressing the problem

$$(\mathbf{x}_{n,t+1}, m_{n,t+1}) = \arg\max_{\substack{\mathbf{x}\in\mathcal{X} \\ m\in\mathcal{M}}} \frac{I(f_n^*, \boldsymbol{\theta}; y_n^{(m)}|\mathbf{x}, \mathcal{D}_{n-1}, \mathcal{D}_{n,t})}{[\lambda^{(m)}]^{\omega}}, \tag{5.30}$$

which is evaluated with respect to the joint distribution

$$\begin{aligned} &p(f_n^*, f_n^{(m)}(\mathbf{x}), \boldsymbol{\theta}, y_n^{(m)}|\mathbf{x}, \mathcal{D}_{n-1}, \mathcal{D}_{n,t}) \\ &= p(\boldsymbol{\theta}|\mathcal{D}_{n-1}, \mathcal{D}_{n,t}) p(f_n^*, f_n^{(m)}(\mathbf{x})|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t}) p(y_n^{(m)}|f_n^{(m)}(\mathbf{x})), \end{aligned} \tag{5.31}$$

where the distribution $p(\boldsymbol{\theta}|\mathcal{D}_{n-1}, \mathcal{D}_{n,t})$ is the posterior distribution on the shared model parameters $\boldsymbol{\theta}$. This distribution represents the transferable knowledge extracted from all the available observations at round t for task n.

Using the chain rule for mutual information [27], the acquisition function (5.30) is expressed as

$$\begin{aligned} \alpha^{\text{MFT-MES}}(\mathbf{x}, m) = &\frac{1}{[\lambda^{(m)}]^{\omega}} \Big[\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D}_{n-1}, \mathcal{D}_{n,t})}\left[I(f_n^*; y_n^{(m)}|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t})\right] \\ &+ I(\boldsymbol{\theta}; y_n^{(m)}|\mathbf{x}, \mathcal{D}_{n-1}, \mathcal{D}_{n,t})\Big]. \end{aligned} \tag{5.32}$$

In (5.32), the first term is the expected information gain on the global optimum f_n^* that is also included in the MF-MES acquisition function in (5.21), while the second term in (5.32) quantifies transferable knowledge via the information gain about the shared parameters $\boldsymbol{\theta}$.

The second term in (5.32) can be written more explicitly as

$$\begin{aligned} &I(\boldsymbol{\theta}; y_n^{(m)}|\mathbf{x}, \mathcal{D}_{n-1}, \mathcal{D}_{n,t}) \\ &= H(y_n^{(m)}|\mathbf{x}, \mathcal{D}_{n-1}, \mathcal{D}_{n,t}) - \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D}_{n-1}, \mathcal{D}_{n,t})}\left[H(y_n^{(m)}|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t})\right] \\ &= H\left(\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D}_{n-1}, \mathcal{D}_{n,t})}\left[p(y_n^{(m)}|\mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_{n,t})\right]\right) - \frac{1}{2}\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D}_{n-1}, \mathcal{D}_{n,t})}\left[\log\left(2\pi e\left[\sigma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x})\right]^2\right)\right], \end{aligned} \tag{5.33}$$

where the first term in (5.33) is the differential entropy of a mixture of Gaussians. In fact, each distribution $p(y_n^{(m)}|\mathbf{x},\boldsymbol{\theta},\mathcal{D}_{n,t})$ corresponds to the GP posterior with mean $\mu_{\boldsymbol{\theta}}^{(m)}(\mathbf{x})$ and variance $[\sigma_{\boldsymbol{\theta}}^{(m)}(\mathbf{x})]^2$ introduced in (5.19) and (5.20), respectively.

Evaluating the first term in (5.33) is problematic [27], since the differential entropy of heterogeneous Gaussian mixtures have no closed-form expression, and here we resort to the upper bound obtained via the principle of maximum entropy [190, 123, 129]

$$H\bigg(\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D}_{n-1},\mathcal{D}_{n,t})}\Big[\mathcal{N}(y_n^{(m)}|\mu_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}),\sigma_{\boldsymbol{\theta}}^{(m)2}(\mathbf{x}))\Big]\bigg) \leq \frac{1}{2}\log\Big(2\pi e\mathrm{Var}(y_n^{(m)})\Big), \quad (5.34)$$

where $\mathrm{Var}(y_n^{(m)})$ represents the variance of random variable $y_n^{(m)}$ following the mixture of Gaussian distribution $\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D}_{n-1},\mathcal{D}_{n,t})}[\mathcal{N}(y_n^{(m)}|\mu_{\boldsymbol{\theta}}^{(m)}(\mathbf{x}),\sigma_{\boldsymbol{\theta}}^{(m)2}(\mathbf{x}))]$. The tightness of upper bound (5.34) depends on the diversity of the Gaussian components with respect to the distribution $p(\boldsymbol{\theta}|\mathcal{D}_{n-1},\mathcal{D}_{n,t})$, becoming more accurate when the components of the mixture tend to the same Gaussian distribution.

When the distribution $p(\boldsymbol{\theta}|\mathcal{D}_{n-1},\mathcal{D}_{n,t})$ is represented using V particles $\{\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_V\}$, the variance in (5.34) can be approximated via the asymptotically consistent estimate [129]

$$\mathrm{Var}(y_n^{(m)}) = \sum_{v=1}^{V}\frac{1}{V}\big([\sigma_{\boldsymbol{\theta}_v}^{(m)}(\mathbf{x})]^2+[\mu_{\boldsymbol{\theta}_v}^{(m)}(\mathbf{x})]^2\big) - \left(\frac{1}{V}\sum_{v=1}^{V}\mu_{\boldsymbol{\theta}_v}^{(m)}(\mathbf{x})\right)^2 + \sigma^2. \quad (5.35)$$

Consequently, by plugging (5.34) into (5.33), the second term in (5.32) is replaced by the quantity

$$\begin{aligned}\Delta\alpha^{\text{MFT-MES}}(\mathbf{x},m) =& \frac{1}{2}\Bigg\{\log\Bigg(2\pi e\bigg(\frac{1}{V}\sum_{v=1}^{V}\big([\sigma_{\boldsymbol{\theta}_v}^{(m)}(\mathbf{x})]^2+[\mu_{\boldsymbol{\theta}_v}^{(m)}(\mathbf{x})]^2\big)\\ &-\Big(\frac{1}{V}\sum_{v=1}^{V}\mu_{\boldsymbol{\theta}_v}^{(m)}(\mathbf{x})\Big)^2\bigg)\Bigg)-\frac{1}{V}\sum_{v=1}^{V}\log\Big(2\pi e\big[\sigma_{\boldsymbol{\theta}_v}^{(m)}(\mathbf{x})\big]^2\Big)\Bigg\}. \quad (5.36)\end{aligned}$$

Based on (5.36), the proposed MFT-MES selects the next pair $(\mathbf{x}_{n,t+1},m_{n,t+1})$ for the current task n by maximizing the criterion (5.27) with (5.36), i.e.,

$$(\mathbf{x}_{n,t+1},m_{n,t+1}) = \arg\max_{\substack{\mathbf{x}\in\mathcal{X}\\ m\in\mathcal{M}}}\frac{1}{[\lambda^{(m)}]^{\omega}}\Big[\alpha^{\text{MF-MES}}(\mathbf{x},m)+\beta\Delta\alpha^{\text{MFT-MES}}(\mathbf{x},m)\Big], \quad (5.37)$$

where the scaling parameter $\beta \geq 0$ determines the relative weight assigned to the task of knowledge transfer for future tasks.

5.5.2 Bayesian Learning for the Kernel Parameter Vector

As explained in the previous section, MFT-MES models the shared parameter vector $\boldsymbol{\theta}$ as a random quantity jointly distributed with the MFGP posterior and observation model (5.6),

Algorithm 6: Multi-Fidelity Transferable Max-Value Entropy Search (MFT-MES)

```
Input : Prior $p(\boldsymbol{\theta})$, scaling parameter $\beta$, simulation costs $\{\lambda^{(m)}\}_{m=1}^{M}$, query budget
        $\Lambda$, stepsize $\eta$, number of SVGD iterations $R$
Output : Optimized solution $\mathbf{x}^{\text{opt}}$
Initialize $t=0$, $r=0$, observation dataset $\mathcal{D}_{n,t}=\emptyset$ and $\Lambda^0=\Lambda$
if $n=1$ then
    Generate particles $\{\boldsymbol{\theta}_v\}_{v=1}^{V}$ i.i.d. from the prior $p(\boldsymbol{\theta})$
end
else
    Set particles $\{\boldsymbol{\theta}_v\}_{v=1}^{V}$ to the particles $\{\boldsymbol{\theta}_v^{(R)}\}_{v=1}^{V}$ produced from the previous
      task $n-1$
end
while $\Lambda^t > 0$ do
    for $v \leq V$ do
        Obtain max-value set $\mathcal{F}$ from distribution $p(f_n^*|\mathbf{x},\boldsymbol{\theta}_v,\mathcal{D}_{n,t})$ using Gumbel
          sampling
    end
    Evaluate the transferable knowledge criterion $\Delta\alpha^{\text{MFT-MES}}(\mathbf{x},m)$ using (5.36)
    Obtain the next decision pair $(\mathbf{x}_{n,t+1},m_{n,t+1})$ via (5.37)
    Observe $y_{n,t+1}^{(m)}$ in (5.6) and update observation history
      $\mathcal{D}_{n,t+1}=\mathcal{D}_{n,t}\cup(\mathbf{x}_{n,t+1},m_{n,t+1},y_{n,t+1}^{(m_{n,t+1})})$
    Update the MFGP posterior as in (5.19) and (5.20)
    Calculate remaining budget $\Lambda^{(t+1)}=\Lambda^{(t)}-\lambda^{(m_{n,t+1})}$
    Set iteration $T_n=t$ and $t=t+1$
end
Evaluate the negative marginal log-likelihood $\ell(\boldsymbol{\theta}_v|\mathcal{D}_{n,T_n})$ for all particles
  $v=1,...,V$ using (5.28)
for $r \leq R$ do
    Evaluate the function $\Omega(\boldsymbol{\theta}_v^{(r)})$ as in (5.39)
    Update each particle via the SVGD update $\boldsymbol{\theta}_v^{(r+1)}=\boldsymbol{\theta}_v^{(r)}+\eta\Omega(\boldsymbol{\theta}_v^{(r)})$
end
Return $\mathbf{x}^{\text{opt}}=\arg\max_{t=1,..,T_n} f_n^{(M)}(\mathbf{x}_{n,t})$
```

which is expressed as in (5.31). Furthermore, the evaluation of the MFT-MES acquisition function (5.37) requires the availability of V particles that are approximately drawn from the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D}_{n-1},\mathcal{D}_{n,t})$. In this subsection, we describe an efficient approach to obtain such particles via SVGD as explained in Sec.3.4.

In SVGD [104], the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D}_{n-1},\mathcal{D}_{n,t})$ is approximated via a set of V particles $\{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,...,\boldsymbol{\theta}_V\}$ that are iteratively transported to minimize the *variational inference* (VI) objective denoted by the KL-divergence $\text{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D}_{n-1},\mathcal{D}_{n,t}))$ over a distribution $q(\boldsymbol{\theta})$ represented via particles $\{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,...,\boldsymbol{\theta}_V\}$. This is done via functional gradient descent in the reproducing kernel Hilbert space (RKHS) which describes the best update direction for particles transportation under KL divergence [6].

Specifically, the SVGD update for the v-th particle at each gradient descent round $r+1$ is expressed as

$$\boldsymbol{\theta}_v^{(r+1)} = \boldsymbol{\theta}_v^{(r)} + \eta\Omega(\boldsymbol{\theta}_v^{(r)}), \tag{5.38}$$

where η is the stepsize, and the function $\Omega(\boldsymbol{\theta}_v^{(r)})$ is defined as

$$\begin{aligned}\Omega(\boldsymbol{\theta}_v^{(r)}) = & \frac{1}{V}\sum_{v'=1}^{V}\Big[\tilde{k}(\boldsymbol{\theta}_{v'}^{(r)},\boldsymbol{\theta}_v^{(r)})\underbrace{\nabla_{\boldsymbol{\theta}_{v'}^{(r)}}\Big(\ell(\boldsymbol{\theta}_{v'}^{(r)}|\mathcal{D}_{n,t}) + \log p_n(\boldsymbol{\theta}_{v'}^{(r)})\Big)}_{\text{log-loss gradient}} \\ & + \underbrace{\nabla_{\boldsymbol{\theta}_{v'}^{(r)}}\tilde{k}(\boldsymbol{\theta}_{v'}^{(r)},\boldsymbol{\theta}_v^{(r)})}_{\text{repulsive force}}\Big],\end{aligned} \tag{5.39}$$

while $\tilde{k}(\cdot,\cdot)$ is a kernel function, the loss function $\ell(\boldsymbol{\theta}_{v'}^{(r)}|\mathcal{D}_{n,t})$ is as in (5.28) and $p_n(\boldsymbol{\theta})$ is the prior distribution at current task n. The first term in (5.39) drives particles $\boldsymbol{\theta}_v$ to asymptotically converge to the MAP solution (5.29), while the second term in (5.39) is a repulsive force that maintains the diversity of particles by minimizing the similarity measure $\tilde{k}(\cdot,\cdot)$. The inclusion of the second term allows the variational distribution $q(\boldsymbol{\theta})$ represented by particles $\{\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_V\}$ to capture the multi-modality of the target posterior $p(\boldsymbol{\theta}|\mathcal{D}_{n-1},\mathcal{D}_{n,t})$, providing a more accurate approximation [156].

The prior $p_n(\boldsymbol{\theta})$ for the current task n is obtained based on the kernel density estimation (KDE) of the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D}_{n-1})$ obtained by using particles $\{\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_V\}$ produced at the end of the previous task $n-1$ [14]. The overall procedure of MFT-MES is summarized in Algorithm 6.

5.6 Experiments

In this section, we empirically evaluate the performance of the proposed MFT-MES on the synthetic benchmark adopted in [174, 125], as well as on two real-world applications, namely radio resource management for wireless cellular systems [112, 208] and gas emission source term estimation [98].

5.6.1 Benchmarks

We consider the following benchmarks in all experiments: 1) MF-MES [125], as described in Algorithm 5; and 2) *Continual MF-MES*, which applies Algorithm 6 with $\beta = 0$, thus not attempting to transfer knowledge from previous tasks to current task. To the best of our knowledge, Continual MF-MES is also considered for the first time in this work. Unlike

MF-MES [125], Continual MF-MES share with MFT-MES the use of SVGD for the update of V particles for the parameter vector $\boldsymbol{\theta}$. The V particles $\{\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_V\}$ are carried from one task to the next, implementing a form of continual learning. Unlike MFT-MES, however, the selection of input-fidelity pairs $(\mathbf{x},m)$ aims solely at improving the current task via the MF-MES acquisition function (5.27), setting $\beta=0$ in (5.37).

The extra computational complexity of Continual MF-MES is brought by the SVGD update in all previous N tasks, i.e., an $\mathcal{O}(NC_{SVGD})$ computation with C_{SVGD} being the complexity of SVGD computation in for a single task. While MFT-MES shares the same computational complexity as Continual MF-MES since the dominant complexity of computing $\Delta\alpha^{\text{MFT-MES}}(\mathbf{x},m)$ in (5.36) brought by extracting $[\sigma_{\boldsymbol{\theta}_v}^{(m)}(\mathbf{x})]^2$ can be mitigated by buffering the values obtained in calculating $\alpha_{\boldsymbol{\theta}}^{\text{MF-MES}}(\mathbf{x},m)$ in (5.27).

5.6.2 Evaluation and Implementation

For all schemes, the parametric mapping $\psi_{\boldsymbol{\theta}}(\cdot)$ in (5.12) was instantiated with a fully-connected neural network consisting of three layers, each with 64 neurons and $tanh$ activation function. Unless stated otherwise, the MFGP model is initialized with $2d+2$ random evaluations across all fidelity levels before performing BO [125], and all the results are averaged over 100 experiments, with figures reporting 90% confidence level. Each experiment corresponds to a random realization of the observation noise signals, random samples of SVGD particles $\{\boldsymbol{\theta}_v\}_{v=1}^V$ at task $n=1$, and random draws of the max-value sets $\mathcal{F}$ in (5.27).

For MF-MES, the parameter vector $\boldsymbol{\theta}$ is set as a random sample from the prior distribution $p(\boldsymbol{\theta})$ for each task. The prior distribution is defined as a zero-mean isotropic Gaussian, i.e., $p(\boldsymbol{\theta})=\mathcal{N}(0,\sigma_p^2\mathbf{I})$ with variance $\sigma_p^2=0.5$ across all the experiments. Furthermore, for the SVGD update (5.39), we used a Gaussian kernel $\tilde{k}(\boldsymbol{\theta},\boldsymbol{\theta}')=\exp(-h||\boldsymbol{\theta}-\boldsymbol{\theta}'||^2)$ with bandwidth parameter $h=1/1.326$. We set $S=10$ in (5.27) and $R=2000$ in Algorithm 6.

5.6.3 Synthetic Optimization Tasks

For the first experiment, we consider synthetic optimization tasks defined by randomly generating Hartmann 6 functions [34]. Accordingly, the input domain is defined as $\mathcal{X}=[0,1]^6$, and the objective value $f_n^{(m)}(\mathbf{x})$ for a task n at fidelity level m is obtained as [125]

$$f_n^{(m)}(\mathbf{x})=-\sum_{i=1}^{4}a_{i,m}\exp\bigg(-\sum_{j=1}^{6}\Delta_{i,j,n}A_{i,j}(x_j-P_{i,j})^2\bigg), \tag{5.40}$$

where $a_{i,m}$, $A_{i,j}$ and $P_{i,j}$ are the (i,m)-th, (i,j)-th, and (i,j)-th entries, respectively, of matrices

$$\mathbf{a} = \begin{pmatrix} 1 & 1.01 & 1.02 & 1.03 \\ 1.2 & 1.19 & 1.18 & 1.17 \\ 3 & 2.9 & 2.8 & 2.7 \\ 3.2 & 3.3 & 3.4 & 3.5 \end{pmatrix}, \mathbf{A} = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}, \tag{5.41}$$

and

$$\mathbf{P} = 10^{-4} \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix}. \tag{5.42}$$

Optimization tasks differ due to the parameter $\Delta_{i,j,n}$ in (5.40) which are generated in an i.i.d. manner from the uniform distribution $\mathcal{U}(0.8, 1.2)$. We set the cost levels as $\lambda^{(1)} = 10, \lambda^{(2)} = 15, \lambda^{(3)} = 20$, and $\lambda^{(4)} = 25$, the total query cost budget in constraint (5.9) to $\Lambda = 500$ for all tasks, and choose the observation noise variance in (5.6) as $\sigma^2 = 0.1$.

We evaluate the performance of all methods at the end of each task via the *simple regret*, which is defined for a task n as [17]

$$\mathrm{SR}_n = f_n^* - \max_{t=1,..,T_n} f_n^{(M)}(\mathbf{x}_{n,t}). \tag{5.43}$$

The simple regret (5.43) describes the error for the best decision $\mathbf{x}_{n,t}$ made throughout the optimization process.

Fig. 5.3 shows the simple regret (5.43) as a function of the number of tasks n observed so far. For MFT-MES, the weight parameter in (5.37) is set to $\beta = 1.2$. Since MF-MES does not attempt to transfer knowledge across tasks, its average performance is constant for all values of n. By transferring knowledge across tasks, both Continual MF-MES and MFT-MES can reduce the simple regret as the number of observed tasks n increases.

MFT-MES outperforms Continual MF-MES as soon as the number of tasks, n, is sufficiently large, here $n > 2$. The advantage of Continual MF-MES for small values of n is due to the fact that MF-MES focuses solely on the current task, while MFT-MES makes decisions also with the goal of improving performance on future tasks. In this sense, the price paid by MFT-MES to collect transferable knowledge is a minor performance degradation for the initial tasks. The benefits of the approach are, however, very significant for later tasks. For instance, at task $n = 10$, MFT-MES decreases the simple regret by a factor of three as compared to Continual MF-MES when the number of particles is $V = 10$.

It is also observed that increasing the number of particles V, is generally beneficial for both Continual MF-MES and MFT-MES. The performance gain with a larger V can

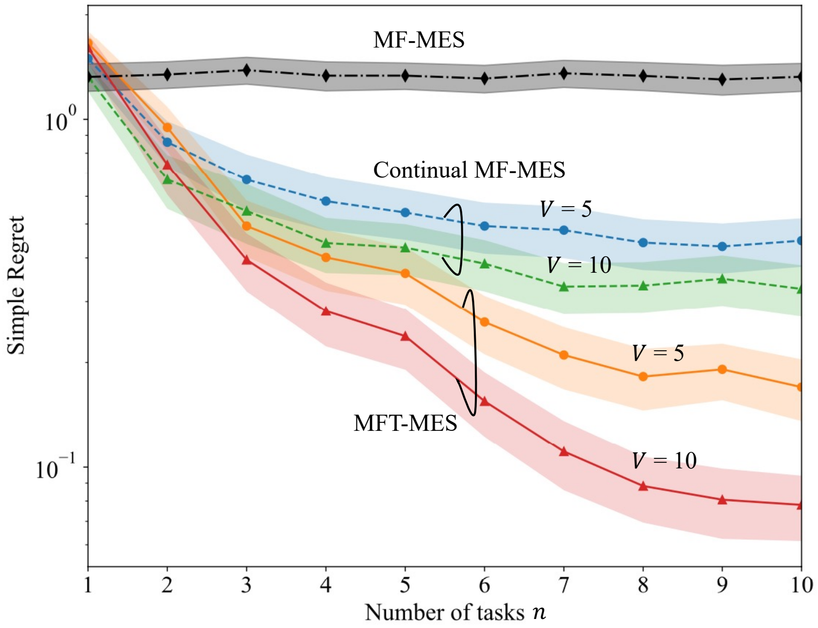

Figure 5.3 Synthetic optimization tasks: Simple regret (5.43) against the number of tasks, n, for MF-MES, Continual MF-MES $(\beta = 0)$ with $V = 5$ and $V = 10$ particles, and MFT-MES ($\beta = 1.2$) with $V = 5$ and $V = 10$ particles.

be ascribed to the larger capacity of retaining information about the uncertainty on the optimized parameter vector $\boldsymbol{\theta}$.

Fig. 5.4 demonstrates the impact of the weight parameter β on the simple regret as a function of the number of tasks. Recall that the performance levels of MF-MES, as well as of Continual MF-MES, do not depend on the value of weight parameter β, which is an internal parameter of the MFT-MES acquisition function (5.37). The optimal value of β, which minimizes the simple regret, is marked with a star. It is observed that it is generally preferable to increase the value of β as the number of tasks n grows larger. This is because a larger β favors the selection of input and fidelity level that focus on the performance of future tasks, and this provident approach is more beneficial for longer time horizons. For example, when the sequence of tasks is short, i.e., when $n \leq 2$, the choice $\beta = 0$, i.e., Continual MF-MES attains best performance; while the weight parameter $\beta = 1.2$ produces best performance given $n = 9$ tasks in the sequence.

5.6.4 Radio Resource Management

In this section, we study an application to wireless communications presented in [112, 208]. Note that, other wireless communication models in the form of sequential expensive-to-evaluate black-box optimization problems can also be deployed with MFT-MES. The problem involves optimizing parameters x that dictate the power allocation strategy of base stations in a cellular system. The vector x contains two parameters for each base station

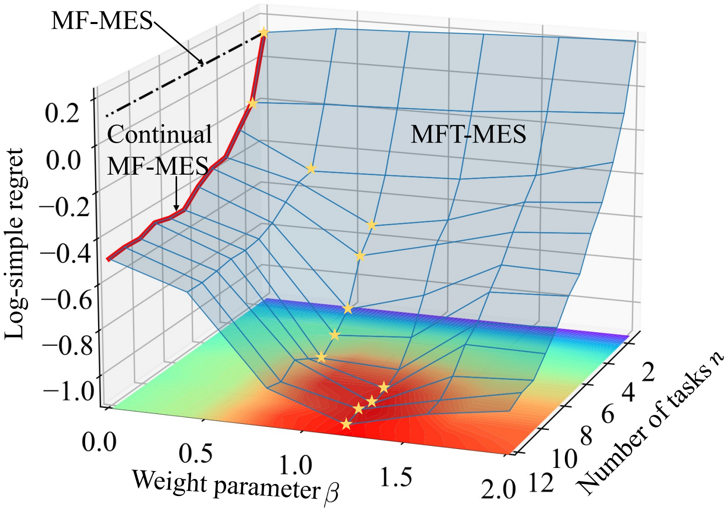

Figure 5.4 Synthetic optimization tasks: Log-simple regret against weight parameter β and the number of tasks, n, for MF-MES (black dash-dotted line), Continual MF-MES (red solid line), and MFT-MES (blue surface). The optimal values of β at the corresponding number of tasks n are labeled as gold stars. The number of particles for Continual MF-MES and MFT-MES is set to $V = 10$.

with the first parameter taking 114 possible values and the second parameter taking 8 possible values. Tasks are generated by randomly deploying users in a given geographical area of radius up to 200 meters, while restricting the users' locations to change by no more than 10 meters for task n as compared to the most recent task $n-1$.

The objective function $f_n(\mathbf{x})$ is the sum-spectral efficiency at which users transmit to the base stations. Evaluating the sum-spectral efficiency requires averaging out the randomness of the propagation channels, and the simulation costs determines the number of channel samples used to evaluate this average. Accordingly, we set the cost levels as $\lambda^{(1)} = 10, \lambda^{(2)} = 20, \lambda^{(3)} = 50$, and $\lambda^{(4)} = 100$, which measure the number of channel samples used at each fidelity level. The total query cost budget is set to $\Lambda = 2000$ for every task.

We initialize the MFGP model with 10 random evaluations across all fidelity levels, and the observation noise variance is $\sigma^2 = 0.83$. In a manner consistent with [208], the performance of each method is measured by the *optimality ratio*

$$\max_{t=1,..,T_n} \frac{f_n^{(M)}(\mathbf{x}_{n,t})}{f_n^*}, \tag{5.44}$$

which evaluates the best function of the optimal value f_n^* attained during the optimization process. Finally, we set $V = 10$ particles for Continual MF-MES and MFT-MES. All other experimental settings are kept the same as in Sec. 5.6.2.

In Fig. 5.5, we set the weight parameter $\beta = 1.6$ for MFT-MES and plot the optimality ratio (5.44) as a function of the number of tasks, n. Confirming the discussions in Sec.

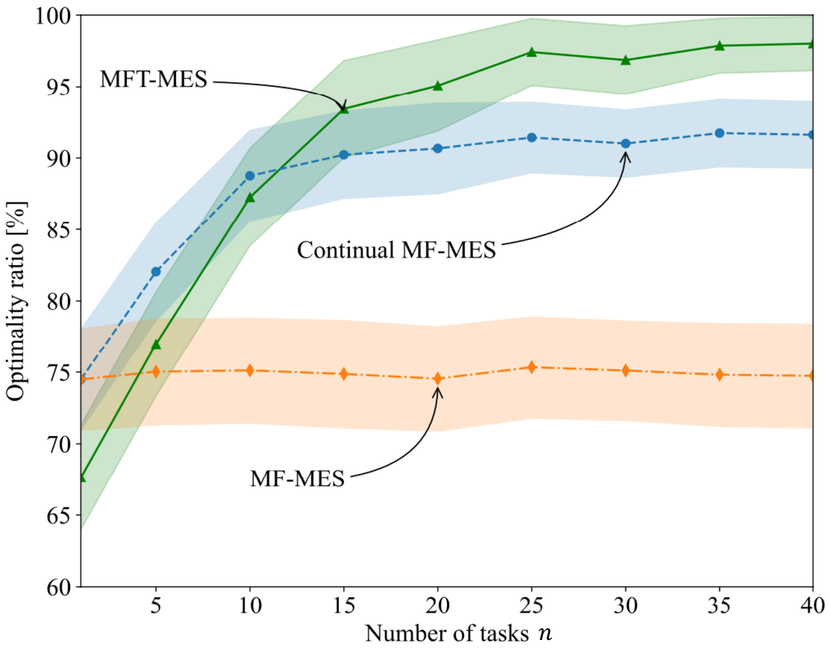

Figure 5.5 Radio resource management for wireless systems [208]: Optimality ratio (5.44) against the number of tasks, n, for MF-MES, Continual MF-MES ($\beta = 0$), and MFT-MES ($\beta = 1.6$) with $V = 10$ particles.

5.6.3, the performance of MF-MES is limited by the lack of information transfer across tasks. In contrast, the performance of both Continual MF-MES and MFT-MES benefits from information transfer. Furthermore, MFT-MES outperforms Continual MF-MES after processing 12 tasks. Through a judicious choice of input and fidelity levels targeting the shared parameters $\boldsymbol{\theta}$, at the end of 40-th task, MFT-MES provides, approximately, an 7% gain in terms of optimality ratio over Continual MF-MES, and a 23% gain over MF-MES.

The impact of the weight parameter β used by MFT-MES on the performance evaluated at the 20-th and 40-th task is illustrated in Fig. 5.6. MFT-MES with any weight parameter $\beta > 0$ outperforms all other schemes given the same number of tasks observed so far. The best performance of MFT-MES is obtained for $\beta = 1.2$ at $n = 20$ tasks, and for $\beta = 1.6$ at $n = 40$ tasks. Therefore, as discussed in Sec. 5.6.3, a larger value of weight parameter β is preferable as the number of tasks n increases. Moreover, the performance of MFT-MES is quite robust to the choice of β. For instance, selecting larger weights β, up to $\beta = 2.4$, is seen to yield a mild performance degradation as compared to the best settings of weight parameters $\beta = 1.2$ for $n = 20$ and $\beta = 1.6$ for $n = 40$. As stated in Sec. 3.5, the design guideline of weight parameter β may adopt multiple rounds of cross-validation over partitions of data samples from each task in practice.

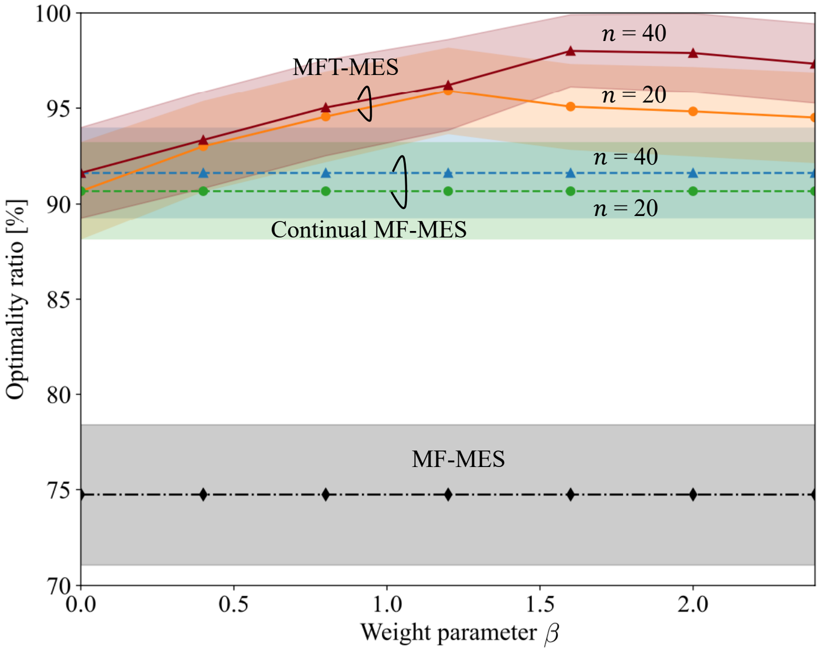

Figure 5.6 Radio resource management for wireless systems [208]: Optimality ratio (5.44) against weight parameter β, for MF-MES, Continual MF-MES with $n = 20$ and $n = 40$ tasks observed, and MFT-MES with $n = 20$ and $n = 40$ tasks observed. The number of particles for Continual MF-MES and MFT-MES is set to $V = 10$.

5.6.5 Gas Emission Source Term Estimation

Finally, we consider an application to the reverse problem formulation of gas emission source term estimation introduced in [98]. The problem aims to optimize a decision vector $\mathbf{x}$ that identifies the characteristics of the gas emission point source based on the Pasquill-Gifford dispersion model [28]. The feasible input domain of the vector is defined as $\mathbf{x} \in [10, 5000] \times [-500, 500]^2 \times [0, 10] \subset \mathbb{R}^4$, with the first parameter being the source emission rate and the rest of the parameters describing the location of the emission source. Tasks are distinguished by the different locations of the sensors used to measure the concentration of emissions.

The objective function $f_n(\mathbf{x})$ is defined as the sum of the squared errors between the concentration measured at the sensors and the concentration calculated by the dispersion model given parameters $\mathbf{x}$. The fidelity of the evaluation of the objective function $f_n(\mathbf{x})$ depends on the atmospheric conditions, which can be classified into $M = 6$ fidelity levels controlled by dispersion coefficients as in [98]. We set the cost levels as $\lambda^{(m)} = 10 + 5 \cdot (m - 1)$; the total query cost budget is set to $\Lambda = 750$ for every task; and the observation noise variance is set to $\sigma^2 = 10^{-3}$. The performance of all methods is measured by the simple regret in (5.43).

In Fig. 5.7, we set the weight parameter in (5.37) to $\beta = 1.5$ for MFT-MES, and plot the simple regret (5.43) as a function of the number of tasks n observed so far. The results demonstrate again the capacity of both Continual MF-MES and MFT-MES to transfer

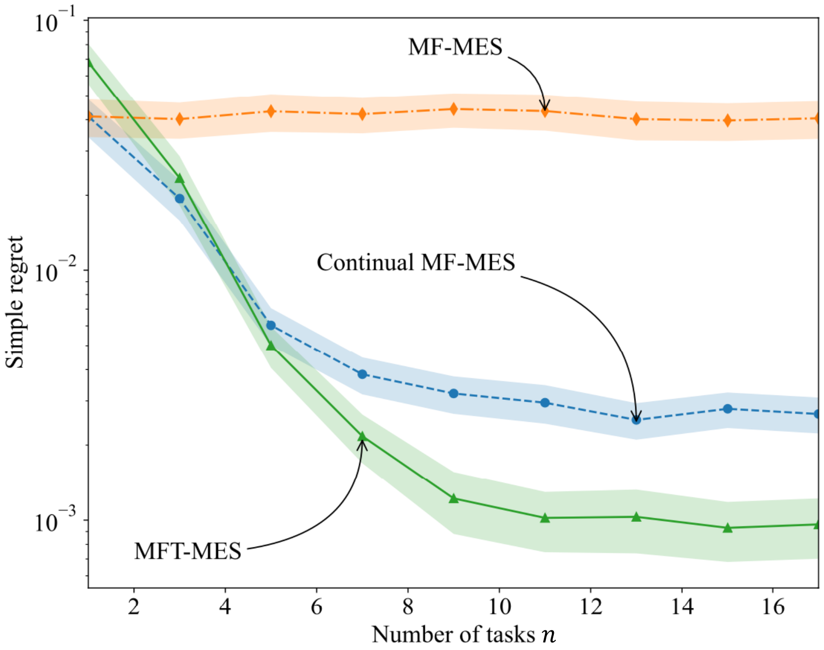

Figure 5.7 Gas emission source term estimation: Simple regret (5.43) against the number of tasks, n, for MF-MES, Continual MF-MES ($\beta = 0$), and MFT-MES ($\beta = 1.5$) with $V = 10$ particles.

knowledge across tasks, achieving better performance as compared to MF-MES. After processing at least four tasks, MFT-MES outperforms Continual MF-MES. In particular, MFT-MES obtains a lower simple regret by a factor of two as compared to Continual MF-MES at the end of task $n = 17$, confirming the importance of accounting for knowledge transfer in the acquisition function (5.37).

In a manner similar to Sec. 5.6.4, we demonstrate the impact of weight parameter β on the simple regret evaluated at the 7-th and 17-th task in Fig. 5.8. The superiority of MFT-MES over all other schemes is observed to hold for any values of weight parameter $\beta > 0$. MFT-MES achieves the best performance with weight parameter around $\beta = 1.0$ for $n = 7$, and approximately $\beta = 1.5$ for $n = 17$. The overall trend confirms the discussion in Sec. 5.6.3, as a larger value of weight parameter β is more desirable when the number of tasks n increases.

5.7 Conclusion

In this chapter, we have have introduced MFT-MES, a novel information-theoretic acquisition function that balances the need to acquire information about the current task with goal of collecting information transferable to future tasks. The key mechanism underlying MFT-MES involves modeling transferable knowledge across tasks via shared inter-task latent variables, which are integrated into the acquisition function design and updated following Bayesian principles. From synthetic optimization tasks and real-world examples,

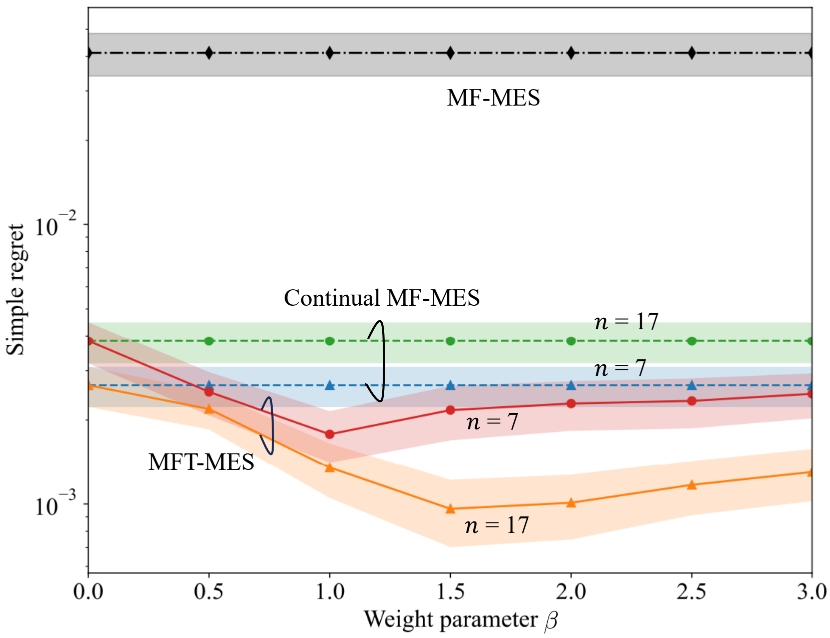

Figure 5.8 Gas emission source term estimation: Simple regret (5.43) against weight parameter β, for MF-MES, Continual MF-MES with $n = 7$ and $n = 17$ tasks observed, and MFT-MES with $n = 7$ and $n = 17$ tasks observed. The number of particles for Continual MF-MES and MFT-MES is set to $V = 10$.

we have demonstrated that the proposed MFT-MES scheme can obtain performance gains as large as an order of magnitude in terms of simple regret as compared to the state-of-art scheme that do not cater to the acquisition of transferable knowledge.

Future work may address theoretical performance guarantees in the forms of regret bounds to explain aspects such as the dependence of the optimal weight parameter β used by MFT-MES as a function of the number of tasks and total query cost budget. Furthermore, it would be interesting to investigate the extensions to multi-objective multi-fidelity optimization problems [97]; the scalability to higher dimensions of the search space [111]; and the potential gains from incorporating generative models [100].

Chapter 6

Bayesian Optimization with Formal Safety Guarantees via Online Conformal Prediction

6.1 Overview

In this Chapter, we focus on achieving reliable BO in the optimization tasks with unknown safety constraint function. Black-box zero-th order optimization is a central primitive for applications in fields as diverse as finance, physics, and engineering. In a common formulation of this problem, a designer sequentially attempts candidate solutions, receiving noisy feedback on the value of each attempt from the system. In this chapter, we study scenarios in which feedback is also provided on the *safety* of the attempted solution, and the optimizer is constrained to limit the number of unsafe solutions that are tried throughout the optimization process. Focusing on methods based on BO, prior art has introduced an optimization scheme – referred to as SAFEOPT – that is guaranteed not to select *any* unsafe solution with a controllable probability over feedback noise as long as strict assumptions on the safety constraint function are met. In this chapter, a novel BO-based approach is introduced that satisfies safety requirements irrespective of properties of the constraint function. This strong theoretical guarantee is obtained at the cost of allowing for an arbitrary, controllable but non-zero, rate of violation of the safety constraint. The proposed method, referred to as SAFE-BOCP, builds on online conformal prediction (CP) and is specialized to the cases in which feedback on the safety constraint is either noiseless or noisy. Experimental results on synthetic and real-world data validate the advantages and flexibility of the proposed SAFE-BOCP.

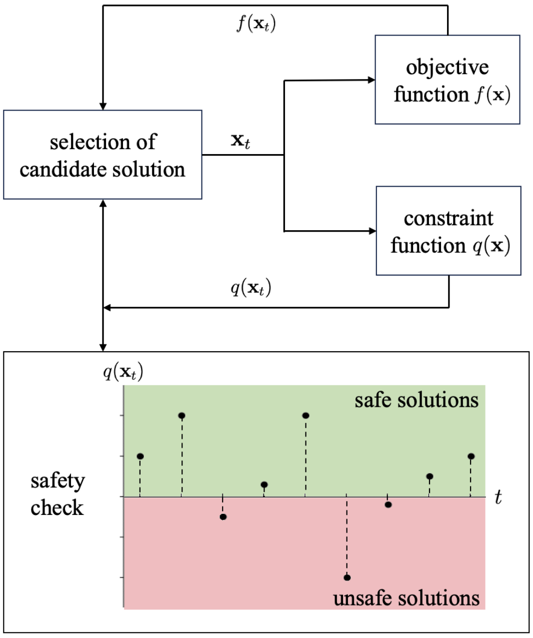

Figure 6.1 This work studies black-box zero-th order optimization with safety constraints. At each step $t = 1, 2, \ldots$ of the sequential optimization process, the optimizer selects a candidate solution $\mathbf{x}_t$ and receives noisy feedback on the values of the objective function $f(\mathbf{x}_t)$ and of the constraint function $q(\mathbf{x}_t)$. Candidate solutions $\mathbf{x}_t$ yielding a negative value for the constraint function, $q(\mathbf{x}_t) < 0$, are deemed to be unsafe. We wish to keep the safety violation rate, i.e., the fraction of unsafe solutions attempted during the optimization process, below some tolerated threshold.

6.2 Introduction

6.2.1 Context and Scope

Problems as diverse as stock portfolio optimization and asset management [119], capacity allocation in energy systems [186], material discovery [194], calibration and optimization of quantum systems [26], and scheduling and optimization of wireless systems [205, 208] can all be formulated as *black-box zero-th order* optimizations. In such problems, the objective to be optimized can only be accessed on individual candidate solutions, and no further information is retrieved apart from the value of the objective. As illustrated in Fig. 6.1, in a common formulation of this problem, a designer sequentially attempts candidate solutions, receiving noisy feedback on the value of each attempt from the system. In this paper, we study scenarios in which feedback is also provided on the *safety* of the attempted solution, and the optimizer is constrained to limit the number of unsafe solutions that are tried throughout the optimization process [167, 13, 179, 168, 146].

As an example, consider the problem of discovering pharmaceuticals for a particular condition (see, e.g., [11]). A pharmaceutical company may try different molecules by carrying out costly trials with patients. Such trials would return not only an indication of

Table 6.1 State of the art on Safe-BO against the proposed Safe-BOCP

	Safe-BO [167, 168, 13, 179, 146]	SAFE-BOCP (ours)
Target safety violation rate	0	(0,1]
Assumption-free safety guarantee	✗	✓

the effectiveness of the candidate cure, but also an indication of possible side effects. A reasonable goal is that of finding a maximally effective compound, while minimizing the number of molecules that are found to have potential side effects during the optimization process.

Typical tools for the solution of black-box zero-th order optimization construct surrogates of the objective function that are updated as information is collected by the optimizer. This can be done using tools from reinforcement learning, such as bandit optimization [158], or Bayesian optimization (BO) [122, 47, 112, 37].

In this chapter, a novel BO-based approach is introduced that satisfies safety requirements irrespective of properties of the constraint function. This guarantee is obtained at the cost of allowing for an arbitrary, controllable but non-zero, rate of violation of the safety constraint. The proposed method, referred to as SAFE-BOCP, builds on online conformal prediction (CP) [55, 40], and is specialized to the cases in which feedback on the safety constraint is either noiseless or noisy.

6.2.2 Related Work

Focusing on methods based on BO, while related works and motivation on safety concern has been detailed in Sec.1.2.2, prior art has introduced an optimization scheme – referred to as SAFEOPT [167, 13] – that is guaranteed not to select *any* unsafe solution with a controllable probability with respect to feedback noise. This theoretical guarantee is, however, only valid if the optimizer has access to information about the constraint function. In particular, reference [167, 13] assumes that the constraint function belongs to a reproducible kernel Hilbert space (RKHS), and that it has a known finite RKHS norm. In practice, specifying such information may be difficult, since the constraint function is a priori unknown.

CP is a general framework for the calibration of statistical models [182]. CP methods can be applied to pre-trained machine learning models with the goal of ensuring that the model's outputs provide reliable estimates of their uncertainty. There are two main classes of CP techniques: *offline CP*, which leverages offline calibration data for this purpose [8, 20, 182]; and *online CP*, which uses feedback on the reliability of past decisions to adjust the post-processing of model's outputs [40, 55]. In both cases, CP offers theoretical guarantees on the quality of the uncertainty quantification provided by the decisions of the system.

The relevance of *online CP* for the problem of interest, illustrated in Fig. 6.1, is that, as the optimizer attempts multiple solutions over time, it needs to maintain an estimate of the constraint function. In order to ensure the safety of the candidate solutions selected by the optimizers, it is important that such estimates come with well-calibrated uncertainty intervals. In this paper, we leverage the theoretical guarantees of online CP in order to define novel BO-based safe optimization strategies.

The only existing combination of CP and BO we are aware of are provided by [163], which apply *offline* CP to BO for the solution of an *unconstrained* optimization problem. The approach aims at improving the acquisition function while accounting for observation noise that goes beyond the standard homoscedastic Gaussian assumption. These prior works do not address safety requirements.

6.2.3 Main Contributions

In this chapter, we introduce SAFE-BOCP, a novel BO-based optimization strategy for constrained black-box zero-th order problems with safety constraints. SAFE-BOCP provides *assumptions-free* guarantees on the safety level of the attempted candidate solutions, while enabling any non-zero target safety violation level. As summarized in Table 6.1, this contrasts with the state-of-the-art papers [167, 168, 13, 179, 146] that only target the most stringent safety constraint with no safety violations throughout the optimization process, while relying on strong assumptions on the constraint function [167, 168, 13, 179].

To summarize, the main contributions of this chapter are as follows:

- We introduce the deterministic SAFE-BOCP (D-SAFE-BOCP) algorithm, which assumes noiseless feedback on the constraint function and targets a flexible safety constraint on the average number of candidate solutions that are found to be unsafe. The approach is based on a novel combination of online CP and Safe-BO methods.

- For the case in which feedback on the constraint function is noisy, we introduce the probabilistic SAFE-BOCP (P-SAFE-BOCP) algorithm, which targets a flexible safety constraint on the *probability* that the average number of candidate solutions that are found to be unsafe exceeds a controllable threshold. The method relies on a "caution-increasing" back-off mechanism that compensates for the uncertainty on the safety feedback received from the system.

- We prove that both D-SAFE-BOCP and P-SAFE-BOCP meet their target safety requirements irrespective of the properties of the constraint function.

- We validate the performance of all the proposed methods and theorems on a synthetic data set and on real-world applications.

The rest of this chapter is organized as follows. Sec. 6.3 formulates the constrained black-box zero-th order problem with safety constraints. The general framework of Safe-BO, as well as the representative, state-of-the-art, algorithm SAFEOPT, are reviewed in Sec. 6.4 and Sec. 6.5, respectively. The proposed SAFE-BOCP methods are introduced in the following sections, with D-SAFE-BOCP presented in Sec. 6.6 and P-SAFE-BOCP described in Sec. 6.7. Experimental results on synthetic dataset are provided in Sec. 6.8, and Sec. 6.9 demonstrates results on real-world applications. Finally, Sec. 6.10 concludes this chapter.

6.3 Problem Formulation

In this section, we describe the constrained black-box zero-th order optimization problems for safety-critical scenarios studied in this work. Then, we introduce the general solution framework of interest in the next section, which is referred to as Safe-BO [167, 168, 13, 179, 146].

6.3.1 Optimization Problem and Safety Constraint

We focus on constrained optimization problems of the form

$$\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \quad \text{s.t.} \quad q(\mathbf{x}) \geq 0, \tag{6.1}$$

where objective function $f(\mathbf{x})$ and constraint function $q(\mathbf{x})$ are real valued; and $\mathcal{X}$ is some specified subset of the d-dimensional vector space $\mathbb{R}^d$. Let f^{opt} denote the maximum value of the problem (6.1), which we assume to be finite. We also assume that the set of optimal solutions, achieving the optimal value f^{opt}, is not empty. We write any optimal solution as $\mathbf{x}^{\text{opt}} \in \mathcal{X}$ with $f^{\text{opt}} = f(\mathbf{x}^{\text{opt}})$. Furthermore, we assume that there is a known, non-empty, set $\mathcal{S}_0 \subset \mathcal{X}$ of safe solutions, i.e.,

$$\mathcal{S}_0 \subseteq \{\mathbf{x} \in \mathcal{X} \colon q(\mathbf{x}) \geq 0\}. \tag{6.2}$$

This subset may be as small as a single safe solution $\mathbf{x}_0$ with $q(\mathbf{x}_0) \geq 0$, i.e., $\mathcal{S}_0 = \{\mathbf{x}_0\}$.

We address the optimization problem (6.1) under the following conditions.

• *Zero-th-order black-box access*: The real-valued objective function $f(\mathbf{x})$ and constraint function $q(\mathbf{x})$ are a priori unknown, and only accessible as zero-th-order black boxes. This implies that, given a candidate solution $\mathbf{x}$, the optimizer can evaluate both functions, obtaining the respective values $f(\mathbf{x})$ and $q(\mathbf{x})$. In practice, the evaluations are often noisy, resulting in the observation of noisy values $\tilde{f}(\mathbf{x})$ and $\tilde{q}(\mathbf{x})$. No other information, such as gradients, is obtained by the optimizer about the functions.

• *Efficient optimization*: The optimizer wishes to minimize the number of accesses to both functions $f(\mathbf{x})$ and $q(\mathbf{x})$, while producing a feasible and close-to-optimal solution

$\mathbf{x}^* \in \mathcal{X}$. That is, we wish for the optimizer to output a vector $\mathbf{x}^* \in \mathcal{X}$ that satisfies the constraint $q(\mathbf{x}^*) \geq 0$, with an objective value $f(\mathbf{x}^*)$ close to the maximum value f^{opt}. The performance of the optimizer can be measured by the *optimality ratio*

$$\Delta f(\mathbf{x}^*) = \frac{f(\mathbf{x}^*)}{f^{\mathrm{opt}}}. \tag{6.3}$$

• *Safety*: Interpreting the inequality $q(\mathbf{x}) \geq 0$ as a safety constraint, we consider choices of the optimization variable $\mathbf{x} \in \mathcal{X}$ that result in a negative value of the constraint function $q(\mathbf{x})$ to be *unsafe*, unless the number of such violations of the constraint are kept below a threshold. Accordingly, we will require that the number of evaluations of the constraint function $q(\mathbf{x})$ that result in a violation of the inequality $q(\mathbf{x}) \geq 0$ to be no larger than a pre-determined value. We will formalize this constraint next by describing the general operation of the optimizer.

6.3.2 Sequential Surrogate-Based Safe Optimization

Starting from a given solution $\mathbf{x}_0 \in \mathcal{S}_0$ (6.2), the optimizer sequentially produces *candidate solutions* $\mathbf{x}_1, ..., \mathbf{x}_T \in \mathcal{X}$ across T *trials* or *iterations*. At each iteration t, the optimizer receives noisy observations of the objective value $f(\mathbf{x}_t)$ as

$$y_t = f(\mathbf{x}_t) + \epsilon_{f,t}, \tag{6.4}$$

as well as a noisy observation of the constraint value $q(\mathbf{x}_t)$ as

$$z_t = q(\mathbf{x}_t) + \epsilon_{q,t}, \tag{6.5}$$

where the observation noise for the objective, $\epsilon_{f,t} \sim \mathcal{N}(0, \sigma_f^2)$, is Gaussian with variance σ_f^2, while the observation noise for the constraint, $\epsilon_{q,t}$, can follow any distribution provided that it has a known upper bound on the one-sided right-tail probability (see Assumption 1 in Sec. 6.7.1 for details).

We focus on optimizers that maintain *surrogate models* of functions $f(\mathbf{x})$ and $q(\mathbf{x})$ in order to select the next iterate. To elaborate, let us write as $\mathcal{O}_t$ the overall history of past iterates $(\mathbf{x}_0, ..., \mathbf{x}_t)$ and past observations $(y_0, z_0, ..., y_t, z_t)$ at the end of the t-th iteration, i.e.,

$$\mathcal{O}_t = (\mathbf{x}_0, ..., \mathbf{x}_t, y_0, ..., y_t, z_0, ..., z_t). \tag{6.6}$$

As we detail in the next section, the optimizer maintains probability distributions $p(f|\mathcal{O}_t)$ and $p(q|\mathcal{O}_t)$ on the functions $f(\mathbf{x})$ and $q(\mathbf{x})$ across all values $\mathbf{x} \in \mathcal{X}$ based on the available information $\mathcal{O}_t$. The distributions $p(f|\mathcal{O}_t)$ and $p(q|\mathcal{O}_t)$ summarize the belief of the optimizer regarding the values of the two functions.

At the next iteration $t+1$, the optimizer leverages the distributions $p(f|\mathcal{O}_t)$ and $p(q|\mathcal{O}_t)$ to obtain iterate $\mathbf{x}_{t+1}$ as follows.
• *Safe set*: Using distribution $p(q|\mathcal{O}_t)$, the optimizer identifies a safe set $\mathcal{S}_{t+1} \subseteq \mathcal{X}$, containing solutions $\mathbf{x} \in \mathcal{X}$ deemed by the optimizer to be safe, i.e., to satisfy the constraint $q(\mathbf{x}) \geq 0$.
• *Acquisition*: Using distributions $p(f|\mathcal{O}_t)$ and $p(q|\mathcal{O}_t)$, the optimizer selects the next iterate $\mathbf{x}_{t+1} \in \mathcal{S}_{t+1}$, with the aim of maximizing the likelihood of obtaining a large, i.e., close to 1, optimality ratio (6.3).

6.3.3 Safety Constraints

We now formalize the safety constraint by distinguishing the cases in which the observations (6.5) of constraint function $q(\mathbf{x})$ are: (*i*) *noiseless*, i.e., we have $z_t = q(\mathbf{x}_t)$ in (6.5) with noise power $\sigma_q^2 = 0$; and *(ii) noisy*, i.e., we have a positive observation noise power $\sigma_q^2 > 0$ in (6.5).

Deterministic Safety Constraint

Noiseless observations of the constraint function values allow the optimizer to keep track of the number of iterates $\mathbf{x}_t$ that result in violations of the non-negativity constraint in problem (6.1). Accordingly, with $\sigma_q^2 = 0$, we impose that the non-negativity constraint $q(\mathbf{x}_t) \geq 0$ be violated no more than a tolerated fraction $\alpha \in [0,1]$ of the T iterations. Specifically, given a *target violation rate* $\alpha \in [0,1]$, this results in the deterministic safety requirement

$$\text{violation-rate}(T) := \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}(q(\mathbf{x}_t) < 0) \leq \alpha, \tag{6.7}$$

where $\mathbb{1}(\cdot)$ is the indicator function, i.e., we have $\mathbb{1}(\text{true}) = 1$ and $\mathbb{1}(\text{false}) = 0$. Therefore, in this first case, we target the maximization of function $f(x)$ subject to the safety constraint (6.7) on the optimization process.

Probabilistic Safety Constraint

In the presence of observation noise on the constraint, i.e., with a positive observation noise power $\sigma_q^2 > 0$, the optimizer cannot guarantee the deterministic constraint (6.7). Rather, targeting problem (6.1), the optimizer can only aim at ensuring that the constraint (6.7) be satisfied with a probability no smaller than a *target reliability level* $1-\delta$, with $\delta \in (0,1]$. This results in the *probabilistic* safety constraint

$$\Pr(\text{violation-rate}(T) \leq \alpha) \geq 1-\delta, \tag{6.8}$$

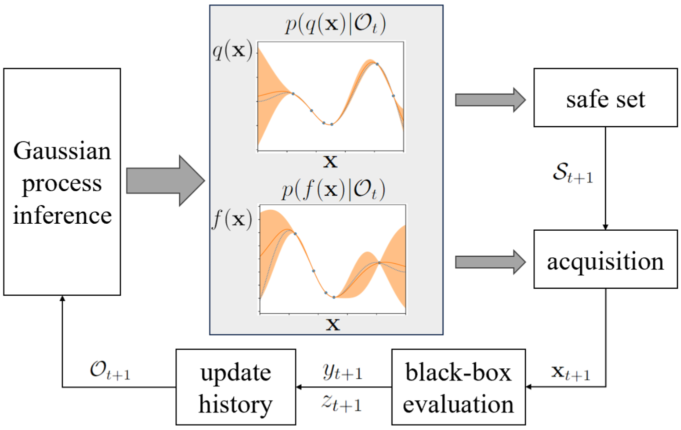

Figure 6.2 Block diagram of Safe-BO schemes consisting of the main steps of safe set creation, producing the safe set $\mathcal{S}_{t+1}$, and of acquisition, selecting the next iterate $\mathbf{x}_{t+1}$.

in which the probability is taken with respect to the observation noise variables $\{\epsilon_{q,t}\}_{t=1}^{T}$ for the constraint function $q(\mathbf{x})$ in (6.5). Therefore, in this second case, we target the maximization of function $f(x)$ subject to the safety constraint (6.8) on the optimization process.

6.4 Safe Bayesian Optimization

We adopt BO as the underlying surrogate-based optimization strategy. When deployed to address the problem of safe black-box optimization defined in the previous section, BO-based schemes are referred to collectively as *Safe-BO* [167, 168, 13, 179, 146]. As illustrated in Fig. 6.2, Safe-BO models objective function $f(\mathbf{x})$ and constraint function $q(\mathbf{x})$ by using independent Gaussian processes (GPs) as surrogate models, producing the distributions $p(f|\mathcal{O}_t)$ and $p(q|\mathcal{O}_t)$ introduced in Sec. 6.3.2. In this section, we adopt the background material on GPs in Sec. 2.1 and discuss a general approach to define safe sets $\mathcal{S}_{t+1}$ on the basis of the current distribution $p(q|\mathcal{O}_t)$.

Let us return to the operation of sequential optimizers based on BO. As explained in the previous section, at the end of iteration t, the optimizer has attempted solutions $(\mathbf{x}_1,...,\mathbf{x}_t)$, which are collectively referred to as $\mathbf{X}_t$. For these inputs, it has observed the noisy values $\mathbf{y}_t = [y_1,...,y_t]^\mathsf{T}$ in (6.4) of the objective function, as well as the noisy values $\mathbf{z}_t = [z_1,...,z_t]^\mathsf{T}$ in (6.5) for the constraint function. As we reviewed in Sec. 2.1, GPs allow the evaluation of the posterior distributions

$$p(f(\mathbf{x})|\mathcal{O}_t) = p(f(\mathbf{x})|\mathbf{X}_t,\mathbf{y}_t) \tag{6.9}$$

and

$$p(q(\mathbf{x})|\mathcal{O}_t) = p(q(\mathbf{x})|\mathbf{X}_t,\mathbf{z}_t) \tag{6.10}$$

for a new candidate solution $\mathbf{x}$, given the history $\mathcal{O}_t = (\mathbf{X}_t, \mathbf{y}_t, \mathbf{z}_t)$ consisted of the previous attempts $\mathbf{X}_t$ and its corresponding noisy observations $\mathbf{y}_t$ and $\mathbf{z}_t$. As we discuss next, these posterior distributions are used by Safe-BO methods to construct *credible intervals*, which quantify the residual uncertainty on the values of functions $f(\mathbf{x})$ and $q(\mathbf{x})$ at any candidate solution $\mathbf{x}$.

Introducing a *scaling parameter* $\beta_{t+1} > 0$, the credible interval for the value of the objective function $f(\mathbf{x})$ for input $\mathbf{x}$ at the end of iteration t, or equivalently at the beginning of iteration $t+1$, is defined by lower bound $f_l(\mathbf{x}|\mathcal{O}_t)$ and upper bound $f_u(\mathbf{x}|\mathcal{O}_t)$ given by

$$\begin{aligned}\mathcal{I}_f(\mathbf{x}|\mathcal{O}_t) &= [f_l(\mathbf{x}|\mathcal{O}_t), f_u(\mathbf{x}|\mathcal{O}_t)]\\ &= [\mu_f(\mathbf{x}|\mathbf{X}_t,\mathbf{y}_t) - \beta_{t+1}\sigma_f(\mathbf{x}|\mathbf{X}_t,\mathbf{y}_t), \mu_f(\mathbf{x}|\mathbf{X}_t,\mathbf{y}_t) + \beta_{t+1}\sigma_f(\mathbf{x}|\mathbf{X}_t,\mathbf{y}_t)],\end{aligned} \tag{6.11}$$

where the mean $\mu_f(\mathbf{x}|\mathbf{X}_t,\mathbf{y}_t)$ and the standard deviation $\sigma_f(\mathbf{x}|\mathbf{X}_t,\mathbf{z}_t)$ are defined as in (2.7) and (2.8), respectively. In a similar manner, the credible interval for the constraint function $q(\mathbf{x})$ is defined as

$$\begin{aligned}\mathcal{I}_q(\mathbf{x}|\mathcal{O}_t) &= [q_l(\mathbf{x}|\mathcal{O}_t), q_u(\mathbf{x}|\mathcal{O}_t)]\\ &= [\mu_q(\mathbf{x}|\mathbf{X}_t,\mathbf{z}_t) - \beta_{t+1}\sigma_q(\mathbf{x}|\mathbf{X}_t,\mathbf{z}_t), \mu_q(\mathbf{x}|\mathbf{X}_t,\mathbf{z}_t) + \beta_{t+1}\sigma_q(\mathbf{x}|\mathbf{X}_t,\mathbf{z}_t)],\end{aligned} \tag{6.12}$$

where the mean $\mu_q(\mathbf{x}|\mathbf{X}_t,\mathbf{y}_t)$ and the standard deviation $\sigma_q(\mathbf{x}|\mathbf{X}_t,\mathbf{z}_t)$ are also defined as in (2.7) and (2.8), respectively.

Under the Gaussian model assumed by GP, the intervals (6.11) and (6.12) include the true function values $f(\mathbf{x})$ and $q(\mathbf{x})$ for a given input $\mathbf{x}$ with probability

$$\mathrm{P}(\beta_{t+1}) = 2F(\beta_{t+1}) - 1, \tag{6.13}$$

where $F(\cdot)$ is the cumulative distribution function (CDF) of standard Gaussian random variable $F(z) = \Pr(Z \leq z)$ with $Z \sim \mathcal{N}(0,1)$. Therefore, the lower bounds $f_l(\mathbf{x}|\mathcal{O}_t)$ and $q_l(\mathbf{x}|\mathcal{O}_t)$ in the credible intervals (6.11) and (6.12), respectively, serve as *pessimistic* estimates of the objective and constraint values at the confidence level defined by probability $\mathrm{P}(\beta_{t+1})$. Furthermore, under the same confidence level, the upper bounds $f_u(\mathbf{x}|\mathcal{O}_t)$ and $q_u(\mathbf{x}|\mathcal{O}_t)$ in (6.11) and (6.12) describe *optimistic* estimates of the objective and constraint values, respectively. That said, it is important to stress that, since the Gaussian model assumed by GP is generally *misspecified*, there is no guarantee on the actual probability that the credible intervals $\mathcal{I}_f(\mathbf{x}|\mathcal{O}_t)$ and $\mathcal{I}_q(\mathbf{x}|\mathcal{O}_t)$ include the true values $f(\mathbf{x})$ and $q(\mathbf{x})$ [80, 9, 193]. These intervals, in fact, are guaranteed to include the true functions values with probability $\mathrm{P}(\beta_{t+1})$ only under the GP model.

In order to meet the safety requirement (6.7) or (6.8), Safe-BO methods define a *safe set* of candidate solutions $\mathbf{x} \in \mathcal{X}$ that are likely to satisfy the constraint $q(\mathbf{x}) \geq 0$.

To this end, the optimizer selects the scaling factor β_{t+1} so as to ensure some desired "safety" probability $\mathrm{P}(\beta_{t+1})$. Then, leveraging the GP model, Safe-BO methods adopt the pessimistic estimate of the value of constraint function given by $q_l(\mathbf{x}|\mathcal{O}_t)$ in (6.12) as a conservative estimate of the constraint function. Accordingly, the safe set $\mathcal{S}_{t+1}$ is defined as the set of all feasible solutions $\mathbf{x} \in \mathcal{X}$ for which the conservative estimate $q_l(\mathbf{x}|\mathcal{O}_t)$ of constraint function $q(\mathbf{x})$ predicts the solution $\mathbf{x}$ to be safe, i.e.,

$$\mathcal{S}_{t+1} = \mathcal{S}(\mathcal{O}_t|\beta_{t+1}) = \{\mathbf{x} \in \mathcal{X} : q_l(\mathbf{x}|\mathcal{O}_t) \geq 0\} \cup \mathcal{S}_0. \tag{6.14}$$

The safe set includes the known initial set $\mathcal{S}_0$ of safe solutions in (6.2), ensuring a non-empty safe set [13].

Safe-BO schemes choose as the first solution $\mathbf{x}_0$ a point randomly selected from the initial safe set $\mathcal{S}_0$. For the following iterations, while all Safe-BO schemes adopt the same definition of the safe set (6.14), the realization of the acquisition process selecting the next iterate $\mathbf{x}_{t+1}$ differentiates the schemes proposed in prior [167, 168, 13, 179, 146]. In the next section, we specifically describe the operation of SAFEOPT [167, 13].

6.5 SAFEOPT

In this section, we review SAFEOPT [167, 13] a representative state-of-the-art Safe-BO method, which will serve as a reference for the proposed SAFE-BOCP strategies introduced in the next section.

6.5.1 Scope and Working Assumptions

SAFEOPT addresses problem (6.1) under a strict version of the probabilistic safety constraint (6.8) with target violation rate $\alpha = 0$ and arbitrary target reliability level $1-\delta$. In order to allow for a zero violation rate ($\alpha = 0$) to be a feasible goal, SAFEOPT makes the assumption that the constraint function $q(\mathbf{x})$ in (6.1) lies in the RKHS $\mathcal{H}_\kappa$ associated with the same kernel function $\kappa(\mathbf{x}, \mathbf{x}')$ assumed by GP inference (see Sec. 2.1). In this sense, the model adopted by GP is assumed by SAFEOPT to be well specified.

Formally, the mentioned assumption made by SAFEOPT enforces that the function can be expressed as

$$q(\mathbf{x}) = \sum_{i=1}^{m} a_i \kappa(\mathbf{x}, \mathbf{x}_i) \tag{6.15}$$

for some vectors $\{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^m$, real coefficients $\{a_i\}_{i=1}^m$, and integer m. For a function $q(\mathbf{x})$ of the form (6.15), the *squared RKHS norm* is defined as

$$||q||_\kappa^2 = \sum_{i=1}^m \sum_{j=1}^m a_i a_j \kappa(\mathbf{x}_i, \mathbf{x}_j). \tag{6.16}$$

Furthermore, a useful property of constraint function $q(\mathbf{x})$ in RKHS $\mathcal{H}_\kappa$ is that it is upper bounded by a function of their squared RKHS norm as

$$|q(\mathbf{x})| \leq \kappa(\mathbf{x}, \mathbf{x})^{1/2} ||q||_\kappa \tag{6.17}$$

for all values $\mathbf{x}$ in their domain. The property (6.17) is leveraged by SAFEOPT by assuming that the RKHS norm of the constraint function $q(\mathbf{x})$ is upper bounded by a known constant B, i.e.,

$$||q||_\kappa \leq B. \tag{6.18}$$

6.5.2 Safe Set Creation

Safe-BO determines the safe set $\mathcal{S}_{t+1}$ in (6.14) using the scaling parameter

$$\beta_{t+1} = B + 4\sigma_q \sqrt{\gamma_t + 1 - \ln(\delta)}, \tag{6.19}$$

where B is the constant appearing in the assumed upper bound (6.18); σ_q^2 is the known observation noise power in (6.5); $1-\delta$ is the target reliability level in (6.8); and γ_t is the *maximal mutual information* between the true values $(q(\mathbf{x}_1), ..., q(\mathbf{x}_t))$ of the constraint function and the corresponding t noisy observations $(\mathbf{z}_1, ..., \mathbf{z}_t)$ when evaluated under the model assumed by GP. This quantity can be evaluated as [13]

$$\gamma_t = \max_{\mathbf{X}'_t = (\mathbf{x}'_1, ..., \mathbf{x}'_t)} \left(\frac{1}{2} \log \left| \mathbf{I}_t + \sigma_q^{-2} \mathbf{K}_q(\mathbf{X}'_t) \right| \right), \tag{6.20}$$

where $\mathbf{I}_t$ is the $t \times t$ identity matrix and $\mathbf{K}_q(\mathbf{X}'_t)$ is the $t \times t$ covariance matrix defined in Sec. 2.1. Evaluating (6.20) requires a maximization over all possible inputs sequences $\mathbf{X}'_t = (\mathbf{x}'_1, ..., \mathbf{x}'_t)$, hence in practice it is often addressed via greedy algorithms (see, e.g., [162]). We also observe that, in the limit of no observation noise, i.e., as $\sigma_q \to 0$, the scaling parameter (6.19) tends to $\beta_t = B$.

By choosing the scaling parameter β_{t+1} as in (6.19), under the key assumption (6.18), all the decisions in the safe set $\mathcal{S}_{t+1}$ (6.14) can be proved to be safe with high probability [13, Lemma 1] (see also [162, Theorem 6]).

6.5.3 Acquisition Process

In this section, we detail the acquisition process adopted by SAFEOPT to select the next iterate $\mathbf{x}_{t+1}$ within the safe set $\mathcal{S}_{t+1}$.

To start, SAFEOPT defines the set of *potential optimizers* $\mathcal{M}_{t+1}$ as the set of all possible solutions $\mathbf{x} \in \mathcal{S}_{t+1}$ that may increase the objective function. It also maintains a set of *possible expanders* $\mathcal{G}_{t+1}$ as the set of safe solutions that can potentially increase the size of the safe set $\mathcal{S}_{t+1}$ if selected. Then, given the potential optimizers $\mathcal{M}_{t+1}$ and the possible expanders $\mathcal{G}_{t+1}$, SAFEOPT chooses the solution $\mathbf{x} \in \mathcal{M}_{t+1} \cup \mathcal{G}_{t+1}$ that maximally reduces the larger uncertainty implied by the credible intervals (6.11) and (6.12), i.e.,

$$\mathbf{x}_{t+1} = \arg \max_{\mathbf{x} \in \mathcal{M}_{t+1} \cup \mathcal{G}_{t+1}} \max\{\sigma_f(\mathbf{x}|\mathcal{O}_t), \sigma_q(\mathbf{x}|\mathcal{O}_t)\}. \tag{6.21}$$

We now describe the construction of sets $\mathcal{M}_{t+1}$ and $\mathcal{G}_{t+1}$. For the first, let us recall that the lower bound $f_l(\mathbf{x}|\mathcal{O}_t)$ in the credible interval (6.11) can be viewed as a pessimistic estimate of the objective $f(\mathbf{x})$, while the upper bound $f_u(\mathbf{x}|\mathcal{O}_t)$ can be interpreted as an optimistic estimate of the same value. The set of potential optimizers, $\mathcal{M}_{t+1}$, includes all safe solutions $\mathbf{x} \in \mathcal{S}_{t+1}$ for which the optimistic estimate $f_u(\mathbf{x}|\mathcal{O}_t)$ is larger than the best pessimistic estimate $f_l(\mathbf{x}|\mathcal{O}_t)$ for all safe solutions $\mathbf{x} \in \mathcal{S}_{t+1}$. This set can be expressed mathematically as

$$\mathcal{M}_{t+1} = \left\{ \mathbf{x} \in \mathcal{S}_{t+1} \middle| f_u(\mathbf{x}|\mathcal{O}_t) \geq \max_{\mathbf{x}' \in \mathcal{S}_{t+1}} f_l(\mathbf{x}'|\mathcal{O}_t) \right\}. \tag{6.22}$$

Note that this set is non-empty, since it includes at least the solution $\mathbf{x}$ that maximizes the lower bound $f_l(\mathbf{x}|\mathcal{O}_t)$.

The set $\mathcal{M}_{t+1}$ accounts only for the objective value to select solutions from the safe set $\mathcal{S}_{t+1}$. In contrast, the set of possible expanders considers the potential impact of a selected candidate solution on the safe set. To formalize this concept, let us write $\mathcal{S}_{t+2}(\mathbf{x})$ for the safe set (6.14) evaluated by extending the current history $\mathcal{O}_t$ with the pair $(\mathbf{x}, q_u(\mathbf{x}|\mathcal{O}_t))$ of candidate solution $\mathbf{x}$ and corresponding hypothetical observation of the optimistic value $q_u(\mathbf{x}|\mathcal{O}_t)$ of the constraint $q(\mathbf{x})$. Accordingly, we have

$$\mathcal{S}_{t+2}(\mathbf{x}) = \mathcal{S}\left(\mathcal{O}_t \cup (\mathbf{x}, q_u(\mathbf{x}|\mathcal{O}_t)) \middle| \beta_{t+1} \right), \tag{6.23}$$

and the set of possible expanders is defined as

$$\mathcal{G}_{t+1} = \{\mathbf{x} \in \mathcal{S}_{t+1} : |\mathcal{S}_{t+2}(\mathbf{x}) \setminus \mathcal{S}_{t+1}| > 0\}, \tag{6.24}$$

that is, as the set of all safe solutions that can potentially increase the size of the safe set.

After T trials, the final decision $\mathbf{x}^*$ is obtained by maximizing the pessimistic estimate $f_l(\mathbf{x}|\mathcal{O}_T)$ of the objective function that is available after the last iteration over the safe set

Algorithm 7: SAFEOPT

```
Input : GP priors $(\mu_f(\mathbf{x}), \kappa_f(\mathbf{x}, \mathbf{x}'))$ and $(\mu_q(\mathbf{x}), \kappa_q(\mathbf{x}, \mathbf{x}'))$, initial safe set $\mathcal{S}_0$,
        initial observation $\mathcal{O}_0$, assumed RKHS norm bound $B$, total number of
        optimization iterations $T$
Output : Decision $\mathbf{x}^*$
Initialize scaling parameters $\{\beta_t\}_{t=1}^{T+1}$ using (6.19), $\mathbf{x}_1 = \mathsf{SAFEOPT}(\mathcal{O}_0|\beta_1)$
for $t = 1, ..., T$ do
    Observe $y_t$ and $z_t$ from candidate solution $\mathbf{x}_t$
    Update the observation history $\mathcal{O}_t = \mathcal{O}_{t-1} \cup \{\mathbf{x}_t, y_t, z_t\}$
    Update GPs with $\mathcal{O}_t$ as in (6.9) and (6.10)
    $\mathbf{x}_{t+1} = \mathsf{SAFEOPT}(\mathcal{O}_t|\beta_{t+1})$
end
Return final decision $\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{S}_{T+1}} f_l(\mathbf{x}|\mathcal{O}_T)$

$\mathsf{SAFEOPT}(\mathcal{O}_t|\beta_{t+1})$:
    Create credible intervals $\mathcal{I}_f(\mathbf{x}|\mathcal{O}_t)$ and $\mathcal{I}_q(\mathbf{x}|\mathcal{O}_t)$ using $\beta_{t+1}$ as in (6.11) and
    (6.12)
    Obtain safe set $\mathcal{S}_{t+1}$ as in (6.14)
    Update the set of potential optimizers $\mathcal{M}_{t+1}$ as in (6.22)
    Update the set of possible expanders $\mathcal{G}_{t+1}$ as in (6.24)
    Return the next iterate $\mathbf{x}_{t+1}$ in accordance to (6.21)
```

$\mathcal{S}_{T+1}$, i.e.,

$$\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathcal{S}_{T+1}} f_l(\mathbf{x}|\mathcal{O}_T). \tag{6.25}$$

The overall procedure of SAFEOPT is summarized in Algorithm 7.

6.5.4 Safety Property

SAFEOPT was shown in [167, 13] to achieve the probabilistic safety constraint (6.8) with $\alpha = 0$, as long as the assumptions that the true constraint function $q(\mathbf{x})$ is of the form (6.15) and that the RKHS norm bound (6.18) holds.

Theorem 1. *(Safety Guarantee of SAFEOPT [13]) Assume that the RKHS norm of the true constraint function $q(\mathbf{x})$ is bounded by $B > 0$ as in* (6.18)*. By choosing the scaling parameter β_{t+1} as in* (6.19)*, SAFEOPT satisfies the probabilistic safety constraint* (6.8) *with $\alpha = 0$. Furthermore, with ideal observations of the constraint function $q(\mathbf{x})$, i.e., $\sigma_q = 0$, by choosing the scaling parameter as $\beta_{t+1} = B$, SAFEOPT meets the deterministic requirement* (6.7) *with $\alpha = 0$.*

From Theorem 1, as long as the Gaussian model assumed by GP is well specified – in the sense indicated by the RKHS form (6.15) with known norm upper bound B in (6.19) – SAFEOPT ensures safe optimization with a zero target violation rate $\alpha = 0$. In practice, however, it is hard to set a value for the constant B. Therefore, for any fixed constant B, the resulting algorithm does not have formal guarantees in terms of safety [146].

6.6 Deterministic Safe-BO via Online CP

As we have reviewed in Sec. 6.5, in order to achieve a zero target violation rate $\alpha = 0$ in the safety constraints (6.7) and (6.8), SAFEOPT assumes that the constraint function $q(\mathbf{x})$ belongs to a specific family of functions. Other Safe-BO methods [168, 13, 179] also require the same assumption to guarantee the safety constraint (see Sec. 6.2). In the following two sections, we will introduce SAFE-BOCP, a novel Safe-BO scheme that achieves the safety constraint requirements (6.7) or (6.8) without requiring *any* assumptions on the underlying constraint function $q(\mathbf{x})$. This goal is met at the cost of obtaining a non-zero, controllable, target violation rate $\alpha \in (0, 1]$ in the deterministic safety requirement (6.7) and in the probabilistic safety requirement (6.8). This section focuses on the case in which observations (6.5) of the constraint function are ideal, i.e., $\sigma_q^2 = 0$, hence aiming at achieving the deterministic safety constraint (6.7). The next section addresses the case with noisy observations on the constraint function.

6.6.1 Adaptive Scaling via Noiseless Feedback on Safety

As detailed in Sec. 6.4, SAFEOPT fixes *a priori* the scaling parameters $\beta_1, ..., \beta_T$ to be used when forming the safe set (6.14), along with the set of potential optimizers (6.22) and possible expanders (6.24), irrespective of the actual history $\mathcal{O}_t$ of past iterates $\mathbf{X}_t$ and observations $\mathbf{y}_t$ and $\mathbf{z}_t$. This is done by leveraging the mentioned assumptions on the constraint function (6.15)–(6.18). In contrast, not relying on any assumption on the constraint function $q(\mathbf{x})$, the proposed SAFE-BOCP selects the scaling parameter β_{t+1} adaptively based on the history $\mathcal{O}_t$ by leveraging ideas from online CP [55, 40].

In order to support the adaptive selection of a scaling parameter β_{t+1} that ensures the deterministic safety constraint (6.7), SAFE-BOCP maintains an *excess violation rate* variable $\Delta\alpha_{t+1}$ across the iterations $t = 1, ..., T$. The variable $\Delta\alpha_{t+1}$ compares the number of previous unsafe candidate solutions $\mathbf{x}'_t$ with $t' = 1, ..., t$ to a tolerable number that depends on the target violation rate α. The main idea is to use the excess violation rate $\Delta\alpha_{t+1}$ to update the parameter β_{t+1}: A larger excess violation rate $\Delta\alpha_{t+1}$ calls for a larger value of β_{t+1} so as to ensure a more pronounced level of pessimism in the evaluation of the safe set (6.14). This forces the acquisition function (6.21) to be more conservative, driving down the excess violation rate towards a desired non-positive value.

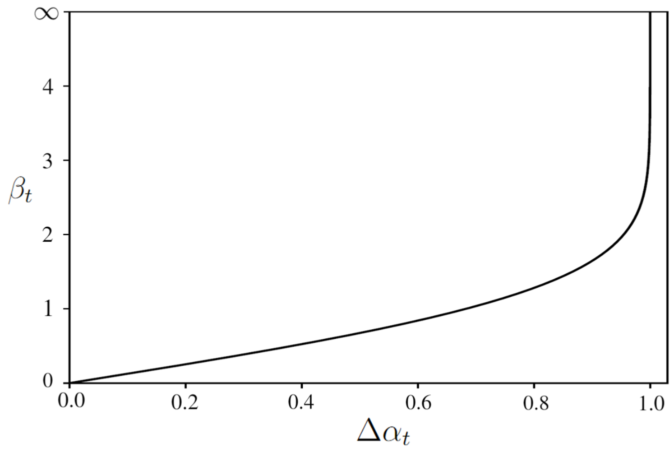

Figure 6.3 Function $\beta_t = \varphi(\Delta\alpha_t)$ in (6.31), which determines the scaling factor β_t as a function of the excess violation rate $\Delta\alpha_t$.

6.6.2 D-SAFE-BOCP

To define the excess violation rate, we first introduce the *safety error signal*

$$\mathrm{err}_t = \mathbb{1}(z_t < 0), \tag{6.26}$$

which yields $\mathrm{err}_t = 1$ if the last iterate $\mathbf{x}_t$ was found to be unsafe based on the observation $z_t = q(\mathbf{x}_t)$, and $\mathrm{err}_t = 0$ otherwise. An important property of schemes, like SAFEOPT and D-SAFE-BOCP, that rely on the use of safe sets of the form (6.14) is that one can ensure a zero error signal $\mathrm{err}_t = 0$ by setting $\beta_t = \infty$. In fact, with this maximally cautious selection, the safe set $\mathcal{S}_t$ includes only the initial safe set $\mathcal{S}_0$ in (6.2), which consists exclusively of safe solutions.

The excess violation rate $\Delta\alpha_{t+1}$ measures the extent to which the average number of errors made so far, $\frac{1}{t}\sum_{t'=1}^{t}\mathrm{err}_{t'}$, exceeds an algorithmic target level α_{algo}, which will be specified later. Accordingly, the excess violation rate is updated as

$$\Delta\alpha_{t+1} = \Delta\alpha_t + \eta(\mathrm{err}_t - \alpha_{\mathrm{algo}}), \tag{6.27}$$

for a given update rate $\eta > 0$ and for any initialization $\Delta\alpha_1 < 1$. The relation between excess violation rate and the average number of errors becomes apparent by rewriting (6.27) as

$$\begin{aligned}\Delta\alpha_{t+1} &= \Delta\alpha_1 + \eta\cdot\left(\sum_{t'=1}^{t}\mathrm{err}_{t'} - \alpha_{\mathrm{algo}}\cdot t\right)\\ &= \Delta\alpha_1 + \eta\cdot t\cdot\Big(\text{violation-rate}(t) - \alpha_{\mathrm{algo}}\Big),\end{aligned} \tag{6.28}$$

which is a linear function of the difference between the violation rate up to time t and the algorithmic target α_{algo}. This implies that the desired safety requirement (6.7) can be equivalently imposed via the inequality

$$\text{violation-rate}(T) = \frac{\Delta\alpha_{T+1} - \Delta\alpha_1}{T\eta} + \alpha_{\text{algo}} \leq \alpha. \tag{6.29}$$

Therefore, controlling the violation rate requires us to make sure that the excess violation rate $\Delta\alpha_t$ does not grow too quickly with the iteration index t.

Intuitively, as mentioned, in order to control the value of the excess violation rate $\Delta\alpha_t$, we need to select values of β_t that increase with $\Delta\alpha_t$. To this end, as summarized in Algorithm 8, inspired by the approach introduced by [40] in the context of online CP, the proposed D-SAFE-BOCP sets the parameter β_t as

$$\beta_t = \varphi(\Delta\alpha_t), \tag{6.30}$$

where we have defined function

$$\varphi(\Delta\alpha_t) = F^{-1}((\text{clip}(\Delta\alpha_t)+1)/2), \tag{6.31}$$

with $F^{-1}(\cdot)$ being the inverse of the function $F(\cdot)$ (6.13), i.e., the inverse CDF of standard Gaussian distribution, and clip$(\Delta\alpha_t) = \max\{\min\{\Delta\alpha_t, 1\}, 0\}$ being the clipping function. An illustration of the function (6.31) can be found in Fig. 6.3. Furthermore, we set the algorithmic target level as

$$\alpha_{\text{algo}} = \frac{1}{T-1}\left(T\alpha - 1 - \frac{1}{\eta} + \frac{\Delta\alpha_1}{\eta}\right). \tag{6.32}$$

The overall procedure of D-SAFE-BOCP is summarized in Algorithm 8. We next prove that D-SAFE-BOCP meets the reliability requirement (6.29).

6.6.3 Safety Guarantees

D-SAFE-BOCP is guaranteed to meet the deterministic safety constraint (6.7) (or equivalently (6.29)), as summarized in the next theorem.

Theorem 2 (Safety Guarantee of D-SAFE-BOCP)**.** *Under noiseless observations of the constraint function ($\sigma_q^2 = 0$), D-SAFE-BOCP satisfies the deterministic safety constraint* (6.7) *for any pre-determined target violation rate $\alpha \in (0,1]$.*

Proof. Function (6.30) implements the following mechanism: When $\Delta\alpha_t \geq 1$, it returns $\beta_t = \infty$, i.e.,

$$\Delta\alpha_t \geq 1 \Rightarrow \beta_t = \infty. \tag{6.33}$$

Algorithm 8: D-SAFE-BOCP

```
Input : GP priors $(\mu_f(\mathbf{x}), \kappa_f(\mathbf{x},\mathbf{x}'))$ and $(\mu_q(\mathbf{x}), \kappa_q(\mathbf{x},\mathbf{x}'))$, initial safe set $\mathcal{S}_0$,
        initial observation $\mathcal{O}_0$, total number of optimization iterations $T$, target
        violation rate $\alpha$, update rate $\eta > 0$, initial excess violation rate $\Delta\alpha_1 < 1$
Output : Decision $\mathbf{x}^*$
Initialize $\mathbf{x}_1 = \text{SAFEOPT}(\mathcal{O}_0|\beta_1)$ using $\beta_1 = \varphi(\Delta\alpha_1)$ (6.31), algorithmic target
  level $\alpha_{\text{algo}}$ as in (6.32)
for $t = 1,...T$ do
    Observe $y_t$ and $z_t$ from candidate solution $\mathbf{x}_t$
    Update the observation history $\mathcal{O}_t = \mathcal{O}_{t-1} \cup \{\mathbf{x}_t, y_t, z_t\}$
    Update GPs with $\mathcal{O}_t$ as in (6.9) and (6.10)
    Evaluate error signal $\text{err}_t = \mathbb{1}(z_t < 0)$ as in (6.26)
    Update excess violation rate $\Delta\alpha_{t+1} = \Delta\alpha_t + \eta(\text{err}_t - \alpha_{\text{algo}})$ as in (6.27)
    Update scaling parameter $\beta_{t+1} = \varphi(\Delta\alpha_{t+1})$ using (6.31)
    $\mathbf{x}_{t+1}$ =SAFEOPT$(\mathcal{O}_t|\beta_{t+1})$ from Algorithm 7
end
Return final decision $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{S}_{T+1}} f_l(\mathbf{x}|\mathcal{O}_T)$
```

As discussed earlier in this section, this ensures a zero error signal $\text{err}_t = 0$. With this mechanism in place, one can guarantee the upper bound

$$\Delta\alpha_{t+1} < 1 + \eta(1 - \alpha_{\text{algo}}) \tag{6.34}$$

for all $t \geq 1$ given the mentioned initialization $\Delta\alpha_1 < 1$. This is because a value $\Delta\alpha_t \geq 1$ would cause the update term in (6.27) to $-\eta\alpha_{\text{algo}} < 0$, and hence the maximum value is attained when $\Delta\alpha_t$ is approaching, but smaller than, 1, and an unsafe decision is made, causing an update equal to $\eta(1 - \alpha_{\text{algo}})$.

Plugging bound (6.34) back into (6.29), yields the upper bound on the violation rate

$$\text{violation-rate}(T) \leq \frac{1 + \eta(1 - \alpha_{\text{algo}}) - \Delta\alpha_1}{T\eta} + \alpha_{\text{algo}}. \tag{6.35}$$

Therefore, by setting (6.32), we finally verify that the deterministic safety requirement (6.29) is satisfied. □

6.7 Probabilistic SAFE-BOCP

We now turn to the case in which the observations (6.5) of constraint function $q(\mathbf{x})$ are noisy ($\sigma_q^2 > 0$). The main challenge in extending the approach proposed in the previous section is the fact that the error signal (6.26) is an unreliable indication of whether candidate $\mathbf{x}_t$ is safe or not due to the presence of observation noise. Accordingly, we start by proposing an alternative way to measure the excess violation rate.

6.7.1 P-SAFE-BOCP

To proceed, we assume that the observation noise $\epsilon_{q,t}$ in (6.5) has a known upper bound on the right-tail probability $\Pr(\epsilon_{q,t} \geq \omega)$ for all $\omega \in \mathbb{R}$. This basic assumption is also adopted in the robust CP literature [39, Theorem 1]. In Sec. 6.7.3, we will illustrate how to further alleviate this assumption by assuming access to noise samples.

Assumption 1. *The constraint observation noise $\epsilon_{q,t}$, which is independent over $t = 1,...,T$, has a known upper bound $F^{+}(\omega)$ on its one-sided right-tail probability, i.e.,*

$$Pr(\epsilon_{q,t} \geq \omega) \leq F^{+}(\omega) \tag{6.36}$$

for all $t = 1,...,T$ and any $\omega \in \mathbb{R}$.

The main idea underlying the proposed P-SAFE-BOCP is to count as unsafe all solutions $\mathbf{x}_t$ for which the noisy observation $z_t = q(\mathbf{x}_t) + \epsilon_{q,t}$ in (6.5) is smaller than some back-off threshold $\omega_q > 0$. Specifically, we define the safety error signal as

$$\text{err}_t = \mathbb{1}(z_t < \omega_q), \tag{6.37}$$

where the corresponding threshold ω_q is obtained as

$$\omega_q = \inf\{\omega \in \mathbb{R} : F^{+}(\omega) \leq 1-(1-\delta)^{\frac{1}{T}}\}. \tag{6.38}$$

The threshold ω_q increases with the target reliability level $1-\delta$ in the probabilistic safety constraint (6.8). In fact, a larger target reliability level calls for more caution in determining whether a given observation z_t of the constraint function is likely to indicate an unsafe solution or not.

The rationale behind the definitions (6.37)-(6.38) is formalized by the following lemma, which relates the true violation rate (6.7) to the estimated violation rate $\sum_{t=1}^{T} \text{err}_t/T$ using the error signal (6.37).

Lemma 1 (Estimated Violation Rate). *For any iterates $\mathbf{x}_1,...,\mathbf{x}_T$, the true violation rate in* (6.7) *is upper bounded by the accumulated error signal rate in* (6.37) *with probability $1-\delta$, i.e.,*

$$\Pr\left(\textit{violation-rate(T)} \leq \frac{1}{T}\sum_{t=1}^{T} \textit{err}_t\right) \geq (1-F^{+}(\omega))^{T} = 1-\delta, \tag{6.39}$$

in which the probability is taken with respect to the observation noise variables $\{\epsilon_{q,t}\}_{t=1}^{T}$ for the constraint function $q(\mathbf{x})$ in (6.5).

Proof. When a candidate solution $\mathbf{x}_t$ is unsafe, i.e., when $q(\mathbf{x}_t) < 0$, the probability that the error signal err_t in (6.37) correctly reports an error, setting $\text{err}_t = 1$, is lower bounded

Algorithm 9: P-SAFE-BOCP

```
Input : GP priors $(\mu_f(\mathbf{x}), \kappa_f(\mathbf{x},\mathbf{x}'))$ and $(\mu_q(\mathbf{x}), \kappa_q(\mathbf{x},\mathbf{x}'))$, initial safe set $\mathcal{S}_0$,
        initial observation $\mathcal{O}_0$, total number of optimization iterations $T$, target
        violation rate $\alpha$, update rate $\eta > 0$, initial excess violation rate $\Delta\alpha_1 < 1$
Output : Decision $\mathbf{x}^*$
Initialize $\mathbf{x}_1 =$ SAFEOPT$(\mathcal{O}_0|\beta_1)$ using $\beta_1 = \varphi(\Delta\alpha_1)$ (6.31), algorithmic target
  level $\alpha_{\text{algo}}$ as in (6.32)
for $t = 1, \ldots T$ do
    Observe $y_t$ and $z_t$ from candidate solution $\mathbf{x}_t$
    Update the observation history $\mathcal{O}_t = \mathcal{O}_{t-1} \cup \{\mathbf{x}_t, y_t, z_t\}$
    Update GPs with $\mathcal{O}_t$ as in (6.9) and (6.10)
    Evaluate cautious error signal $\text{err}_t = \mathbb{1}(z_t < \omega_q)$ as in (6.37) with $\omega_q$ obtained
      from (6.38)
    Update excess violation rate $\Delta\alpha_{t+1} = \Delta\alpha_t + \eta(\text{err}_t - \alpha_{\text{algo}})$ as in (6.27)
    Update scaling parameter $\beta_{t+1} = \varphi(\Delta\alpha_{t+1})$ using (6.31)
    $\mathbf{x}_{t+1} =$SAFEOPT$(\mathcal{O}_t|\beta_{t+1})$ from Algorithm 7
end
Return final decision $\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{S}_{T+1}} f_l(\mathbf{x}|\mathcal{O}_T)$
```

by $1 - F^+(\omega)$. Therefore, the probability that the true violation rate violation-rate(T) no larger than the estimated violation rate $\sum_{t=1}^{T} \text{err}_t / T = 1$ is lower bounded by the probability that all the errors correctly reported. This is, in turn, lower bounded by $(1 - F^+(\omega))^T$ by the independence of the observation noise variables $\{\epsilon_{q,t}\}_{t=1}^{T}$. □

As specified in Algorithm 9, P-SAFE-BOCP follows the same steps in D-SAFE-BOCP with the caveat that the error signal (6.37) is used in lieu of (6.26). As we prove next, the correction applied via the error signal (6.37) is sufficient to meet the probabilistic safety requirement (6.8).

6.7.2 Safety Guarantees

The satefy guarantees of P-SAFE-BOCP are summarized in the following theorem.

Theorem 3 (Safety Guarantee of P-SAFE-BOCP)**.** *Under noisy observations of the constraint function and Assumption 1, P-SAFE-BOCP satisfies the probabilistic safety constraint* (6.8) *for any pre-determined target violation rate* $\alpha \in (0, 1]$ *and target reliability level* $\delta \in (0, 1)$.

Proof. Using the same arguments as in the proof of Theorem 2, the estimated violation rate can be upper bounded with probability 1 as

$$\frac{1}{T} \sum_{t=1}^{T} \text{err}_t \leq \frac{1 + \eta(1 - \alpha_{\text{algo}}) - \Delta\alpha_1}{T\eta} + \alpha_{\text{algo}}. \tag{6.40}$$

Using this bound with Lemma 1, we conclude that, with probability at least $1-\delta$, in which the probability is taken over the observation noise variables $\{\epsilon_{q,t}\}_{t=1}^{T}$, we have bound on the *true violation rate*

$$\text{violation-rate}(T) \leq \frac{1}{T}\sum_{t=1}^{T}\text{err}_t \leq \alpha, \tag{6.41}$$

which recovers the probabilistic safety constraint (6.8). □

6.7.3 Data-Driven Probability Bound

A possible challenge in applying P-SAFE-BOCP in practice is the fact that an upper bound $F^{+}(\omega)$ on the probability $\text{Pr}(\epsilon_{q,t} \geq \omega)$ may not be known *a priori*. In this subsection, we provide a data-driven approach for evaluating an upper bound on the probability $\text{Pr}(\epsilon_{q,t} \geq \omega)$, assuming only access to i.i.d. observation noise samples.

Lemma 2 (Estimated Upper Bound)**.** *Assume access to i.i.d. observation noise samples $\{\epsilon_{q,i}\}_{i=1}^{m}$. The empirical estimate of the right-tail probability*

$$\hat{F}^{+}(\omega) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{1}(\epsilon_{q,i} > \omega), \tag{6.42}$$

when offset by $\psi > 0$, provides an upper bound on Pr$(\epsilon_{q,i} > \omega)$ with probability

$$\begin{aligned}&\text{Pr}\left(\text{Pr}(\epsilon_{q,i} \geq \omega, \forall \omega \in \mathbb{R}) \leq \hat{F}^{+}(\omega) + \psi\right)\\ &\geq 1-\exp(-2m\psi^2)\end{aligned} \tag{6.43}$$

for any $\psi > \sqrt{\ln 2/2m}$.

Lemma 2 is a direct application of Dvoretsky-Kiefer-Wolfowitz inequality [116].

Consequently, by using $\hat{F}^{+}(\omega)+\psi$ in lieu of $F^{+}(\omega)$ in (6.38), we have the following modified safety guarantee of P-SAFE-BOCP.

Corollary 1. *Under noisy observations of the constraint function, P-SAFE-BOCP with $\hat{F}^{+}(\omega)+\psi$, for any $\psi > 0$, in lieu of $F^{+}(\omega)$ in* (6.38) *satisfies the guarantee*

$$\text{Pr}(\textit{violation-rate}(T) \leq \alpha) \geq (1-\exp(-2m\psi^2))(1-\delta) \tag{6.44}$$

for any pre-determined target violation rate $\alpha \in (0,1]$ and target reliability level $\delta \in (0,1)$, where the probability is taken with respect to the observation noise variables $\{\epsilon_{q,t}\}_{t=1}^{T}$ as well as the m i.i.d. noise samples in (6.42).

Corollary 1 is obtained by combining Lemma 2 and Theorem 3. Intuitively, with an increasing number m of the constraint observation noise samples, the tightness of the

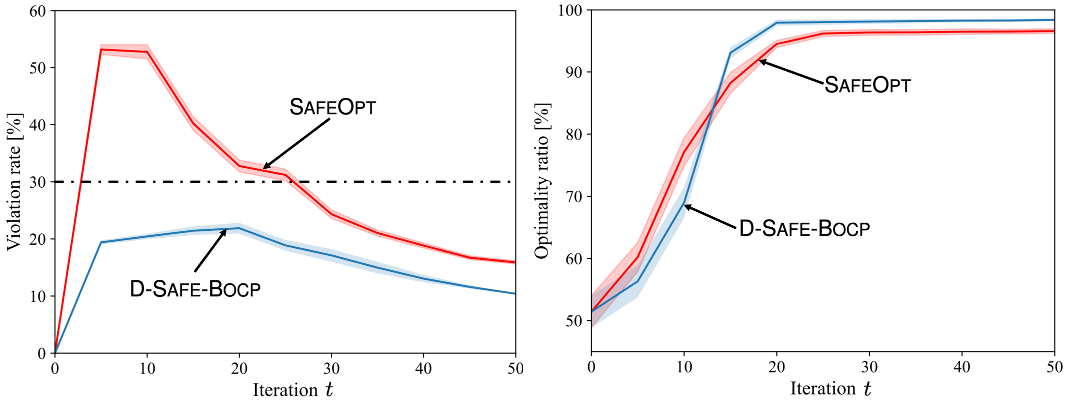

Figure 6.4 Violation-rate(t) (top) and optimality ratio (bottom) against iteration index t with target violation rate $\alpha = 0.3$ (dot-dashed line), update rate $\eta = 2$, misspecified kernel bandwidth $h = 1/14.58$, RKHS norm bound $B = ||q||_{\kappa^*}$ and total number of iteration $T = 50$.

safety guarantee in Theorem 3 is enhanced as a result of increasingly accurate observation noise estimation.

6.8 Numerical Results For a Synthetic Benchmark

In this section, we detail experimental results aimed at comparing SAFE-BOCP with SAFEOPT [13] on a synthetic benchmark inspired by [13].

6.8.1 Synthetic Dataset

In a manner similar to [13], we focus on a synthetic setting with a scalar optimization variable $\mathbf{x} \in \mathbb{R}$ in which the objective function $f(\mathbf{x})$ is a realization of a GP with zero mean and RBF kernel $\kappa^*(\mathbf{x},\mathbf{x}')$ (2.1) with bandwidth $h^* = 1/1.62$, while the constraint function $q(\mathbf{x})$ is a function in this RKHS $\mathcal{H}_{\kappa^*}$ which has the form (6.15) with coefficients $\{a_i\}_{i=0}^{10} = [-0.05, -0.1, 0.3, -0.3, 0.5, 0.5, -0.3, 0.3, -0.1, -0.05]$ and scalars $\{\mathbf{x}_i\}_{i=1}^{10} = [-9.6, -7.4, -5.5, -3.3, -1.1, 1.1, 3.3, 5.5, 7.4, 9.6]$. Accordingly, the constraint function $q(\mathbf{x})$ has RKHS norm $||q||_{\kappa^*} = 1.69$ in (6.18). In order to investigate the impact of misspecification of GP (see Sec. 6.5.1) on Safe-BO including the proposed SAFE-BOCP, we consider the two cases: (*i*) *well-specified GP* that uses $\kappa^*(\mathbf{x},\mathbf{x}')$ for the GP kernel, i.e., $\kappa(\mathbf{x},\mathbf{x}') = \kappa^*(\mathbf{x},\mathbf{x}')$; (*ii*) *misspecified GP* that uses RBF kernel with smaller bandwidth $h = 1/14.58 < h^*$, i.e., $\kappa(\mathbf{x},\mathbf{x}') \neq \kappa^*(\mathbf{x},\mathbf{x}')$, with unknown $||q||_{\kappa}$.

As discussed throughout the paper, the scaling parameter for the constraint function $q(\mathbf{x})$ in (6.12) is *a priori* determined by (6.19) for SAFEOPT, and is *adapted* by feedback via $\beta_{t+1} = \varphi(\Delta\alpha_{t+1})$ (6.27) for the proposed SAFE-BOCP, while we fix the scaling parameter for the objective function $f(\mathbf{x})$ in (6.11) to 3 since it does not affect the safety guarantee for both SAFEOPT (see [162, Theorem 6]) and SAFE-BOCP. The objective observation

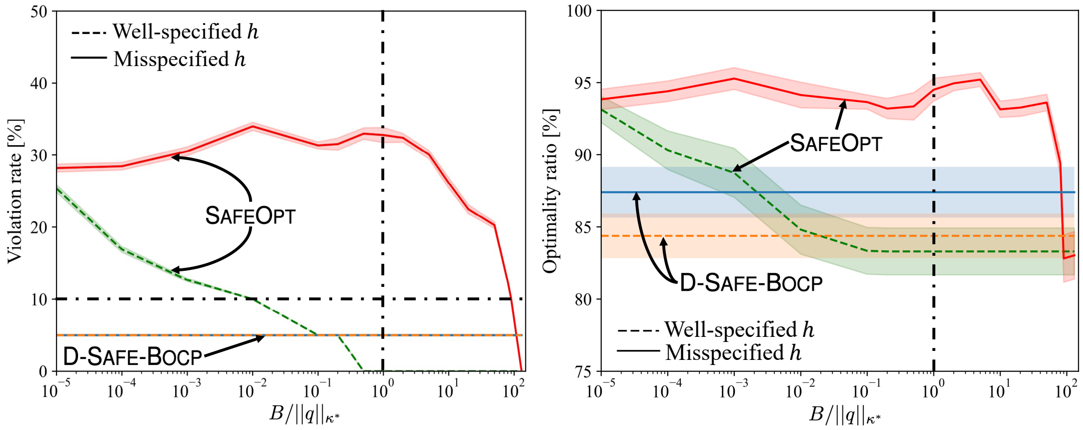

Figure 6.5 Violation rate (6.7) (left) and optimality ratio (6.3) (right) against the ratio between the RKHS norm bound B assumed by GP and the actual norm $||q||_{\kappa^*}$ in (6.18). The dashed lines are obtained with well-specified GP models, which corresponds to kernel bandwidth $h = 1/1.69$ (same one for $\kappa^*(\mathbf{x}, \mathbf{x}')$), while the solid lines are obtained with misspecified GP models, having kernel bandwidth $h = 1/14.58$.

noise variance is set to $\sigma_f^2 = 2.5 \times 10^{-3}$; the initial safe decision is chosen as $\mathbf{x}_0 = 0$ for which we have $q(\mathbf{x}_0) = 0.946 > 0$; and the total number of optimization iterations is set to $T = 25$. For SAFE-BOCP, we set the update rate in (6.27) to $\eta = 2.0$. All results are averaged over 1,000 experiments, with error bars shown to encompass 95% of the realizations. Each experiment corresponds to a random draw of the objective function and to random realization of the observation noise signals.

6.8.2 Deterministic Safety Requirement

As explained in Sec. 6.5, SAFEOPT requires the GP model for the constraint function $q(\mathbf{x})$ to be well specified (6.15)–(6.18) in order to meet safety conditions. To study the impact of violations of this assumption, we start by considering the noiseless case, i.e., $\sigma_q^2 = 0$, and we vary the kernel bandwidth h adopted for the GP models used as surrogates for the objective and constraint functions as discussed earlier.

Fig. 6.4 shows the violation rate and optimality ratio against the iteration index t. For D-SAFE-BOCP, we set the update rate as $\eta = 2$ and the target violation rate to $\alpha = 0.3$, while SAFEOPT assumes target $\alpha = 0$ with RKHS norm bound $B = ||q||_{\kappa^*}$. For both schemes, the total number of iterations is $T = 50$, and the misspecified GP with RBF kernel bandwidth $h = 1/14.58$ is adopted.

The violation rate obtained by SAFEOPT is above the target $\alpha = 0.3$ for a significant interval of time t, and it progressively falls below the target with a larger t, while D-SAFE-BOCP meets the deterministic safety requirement (6.7) with the pre-determined target $\alpha = 0.3$ across all iterations. Furthermore, the optimality ratio obtained by D-SAFE-BOCP is larger than SAFEOPT after iteration $t = 13$, converging to 97.5% at iteration $t = 20$. In

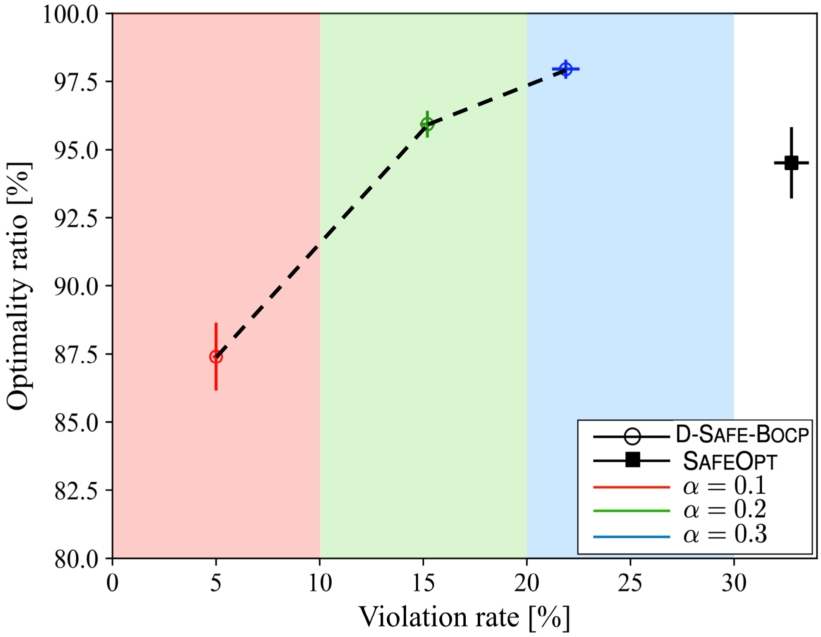

Figure 6.6 Violation-rate(T) and optimality ratio for different target violation rates α for D-SAFE-BOCP, with update rate $\eta = 2$, misspecified kernel bandwidth $h = 1/14.58$, and RKHS norm bound $B = ||q||_{\kappa^*}$. The background colors represent intervals in which the safety requirement (6.7) is met (see text for an explanation).

contrast, SAFEOPT converges to optimality ratio of 94.5% at iteration $t = 25$, at which point the target safety level $\alpha = 0.3$ is violated.

Fig. 6.5 shows the violation-rate(T) in (6.7) with $T = 20$, as well as the optimality ratio (6.3), as a function of constant B assumed by SAFEOPT for both well-specified and misspecified GPs, and the target violation rate is set to $\alpha = 0.1$. Note that the performance of D-SAFE-BOCP does not depend on the value of B, which is an internal parameter for SAFEOPT, but it is affected by the choice of parameter h. By Theorem 1, any value $B \geq ||q||_{\kappa}$ in (6.18) guarantees the safety of SAFEOPT. However, since RKHS norm for the misspecified GP is generally unknown, we plot violation rate and optimality ratio as functions of the ratio $B/||q||_{\kappa^*}$, to highlight the two regimes with well specified and misspecified value of B.

Confirming Theorem 1, with a ratio $B/||q||_{\kappa^*} \geq 1$ for the well-specified GP with kernel $\kappa(\mathbf{x}, \mathbf{x}') = \kappa^*(\mathbf{x}, \mathbf{x}')$, SAFEOPT is seen to strictly satisfy the deterministic safety constraint (6.7), since the violation rate is equal to zero, as per its target. Instead, when $B/||q||_{\kappa^*} < 1$, and/or when the GP is misspecified, i.e., $\kappa(\mathbf{x}, \mathbf{x}') \neq \kappa^*(\mathbf{x}, \mathbf{x}')$, the violation rate exceeds the target α. In contrast, D-SAFE-BOCP obtains a violation rate below the target α, irrespective of kernel bandwidth h assumed in GP.

In terms of optimality ratio, in the regime $B/||q||_{\kappa^*} \geq 1$, with a well-specified GP parameter h, SAFEOPT achieves around 83%, while D-SAFE-BOCP obtains the larger optimality ratio 84.5%. In contrast, with a misspecified value h, D-SAFE-BOCP achieves an optimality ratio around 87.5%, while the optimality ratio of SAFEOPT is larger, but this

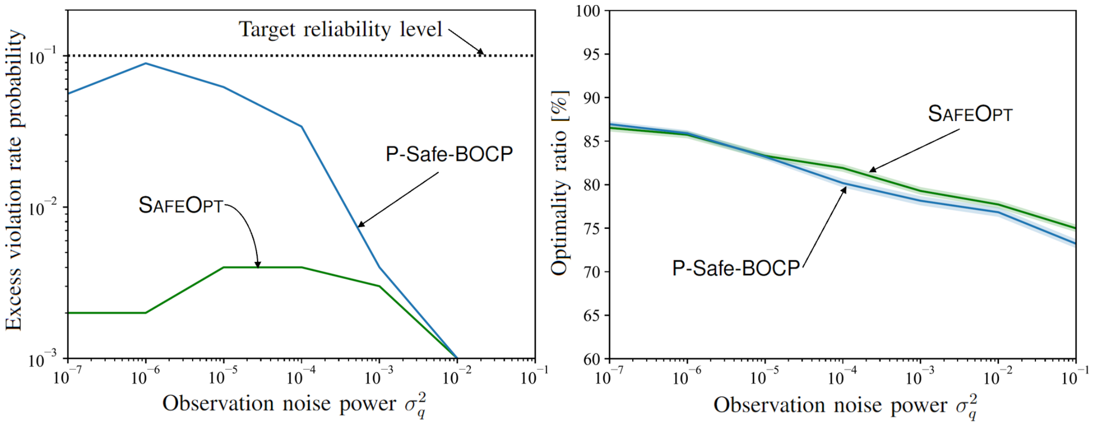

Figure 6.7 Probability of excessive violation rate (6.8) (left) and optimality ratio (6.3) (right) as a function of constraint observation noise power σ_q^2, with update rate $\eta = 2$, RKHS norm bound $B = 10||q||_{\kappa^*}$, and well-specified kernel bandwidth $h = 1/1.62$.

comes at the cost of the violation of the safety requirement. Note that a misspecified value of the kernel bandwidth h does not necessarily reduce the performance of D-SAFE-BOCP, which is improved in this example.

The trade-off between violation rate and optimality ratio is studied in Fig. 6.6 by varying the target violation rate α for D-SAFE-BOCP. For each value of α, we show the achieved pair of violation rate and optimality ratio, along with the corresponding realization ranges along the two axes. Recall that for SAFEOPT the assumed target is $\alpha = 0$, and hence one pair is displayed. We focus here on the misspecified GP case, i.e., $\kappa(\mathbf{x},\mathbf{x}') \neq \kappa^*(\mathbf{x},\mathbf{x}')$, while the SAFEOPT parameter B is selected to the "safe" value $B = ||q||_{\kappa^*}$, which is unaware of kernel misspecification.

For each value of $\alpha \in \{0.1, 0.2, 0.3\}$, the figure highlights the intervals of violation rates that meet the safety requirement (6.7) using different colors. Specifically, for $\alpha = 0.1$, all violation rates below 0.1 are acceptable, as denoted by the red interval; for $\alpha = 0.2$, all violation rates in the red and green intervals are acceptable; and for $\alpha = 0.3$, all violation rates below in the cyan, green, and red interval meet the safety constraint.

The figure shows that the violation rate obtained by SAFEOPT exceeds its target $\alpha = 0$, and thus the safety requirement is violated. In contrast, as per the theory developed in this paper, D-SAFE-BOCP meets violation-rate requirement for all values of the target α. Moreover, as the tolerated violation rate α increases, the optimality ratio of D-SAFE-BOCP is enhanced, indicating a trade-off between the two metrics. When increasing the target violation rate α, D-SAFE-BOCP raises the algorithmic target level α_{algo} in (6.32), making it possible for the optimizer to reduce the time spent under the maximally cautious scaling $\beta_t = \infty$ in (6.30). Consequently, with a larger α, the optimality ratio of D-SAFE-BOCP gains from more explorations of the objective function $f(\mathbf{x})$.

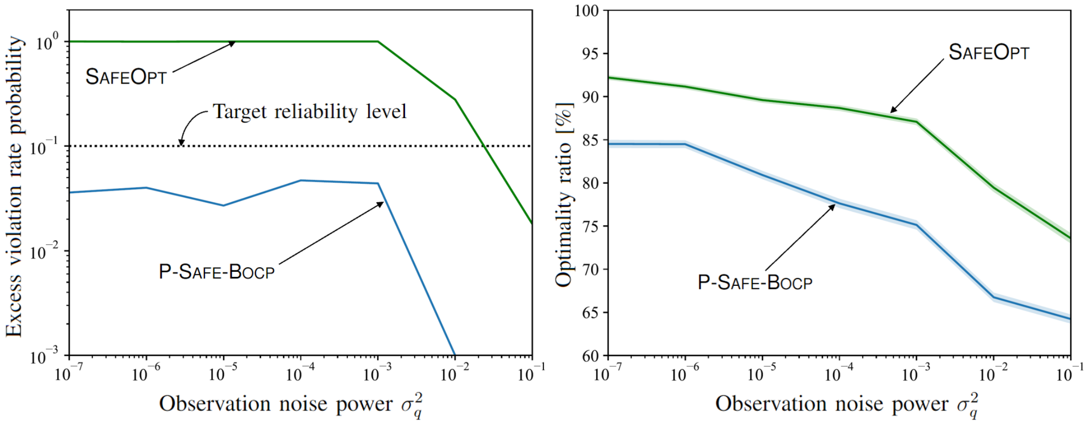

Figure 6.8 Excess violation rate probability (6.8) (left) and optimality ratio (6.3) (right) as a function of constraint observation noise power σ_q^2, with update rate $\eta = 2$, RKHS norm bound $B = 10||q||_{\kappa^*}$, and misspecified kernel bandwidth $h = 1/14.58$.

6.8.3 Probabilistic Safety Constraint

We now turn to considering scenarios with observation noise $\sigma_q^2 > 0$, and aim at evaluating the performance in terms of probabilistic safety requirement (6.8) and optimality ratio (6.3). We set the target reliability level $1-\delta = 0.9$ with target violation rate $\alpha = 0.1$ for P-SAFE-BOCP, and with $\alpha = 0$ for SAFEOPT in accordance with SAFEOPT's design. For the latter scheme, we set the "safe" value $B = 10||q||_{\kappa^*}$, while we consider both the well-specified kernel bandwidth $h = h^* = 1/1.62$, and the misspecified one $h = 1/14.58 < h^*$, as considered also in the previous set of experiments. For all schemes, the excess violation rate probability in (6.8) is obtained by averaging over 10,000 realizations.

We plot the excess violation rate probability (6.8) and the optimality ratio in Fig. 6.7 and Fig. 6.8 against the observation noise power σ_q^2. The first figure corresponds to the case of a well-specified kernel bandwidth while for the second we adopted misspecified value. Confirming the theory, in the former case, both SAFEOPT and P-SAFE-BOCP attain an excess violation rate probability below the target level $1-\delta$. In contrast, for a misspecified kernel, SAFEOPT can only satisfy the constraint (6.8) for sufficiently large observation noise, but P-SAFE-BOCP still meets the probability safety constraint (6.8). We note that a larger observation noise is beneficial to SAFEOPT in terms of safety since it forces a larger level of pessimism in the definition of the safe set $\mathcal{S}_{t+1}$ in (6.14).

In terms of optimality ratio, larger observation noise power σ_q^2 generally yields a degraded optimality ratio. In the well-specified regime considered in Fig. 6.7, both schemes have comparable performance and the optimality ratio gap is no more than 5%. In the misspecified regime demonstrated in Fig. 6.8, the performance levels are not comparable, since the gains of recorded for SAFEOPT come at the cost of violations of the safety constraint (6.8), except for a sufficiently large observation noise power, here $\sigma_q^2 \geq 0.1$.

6.9 Numerical Results for Real World Applications

In this section, we compare SAFEOPT [167] and SAFE-BOCP in two real-world applications, with the goal of validating the safety gains obtained by the proposed method along the optimization process.

6.9.1 Safe Movie Recommendation

As in [167], consider a system that sequentially recommends movies to a user. Each user assigns a score from 1 to 5 to a recommended movie. Following standard matrix factorization algorithms, we introduce a feature vector $\mathbf{x} \in \mathbb{R}^d$ for each movie. Accordingly, selecting a movie amounts to choosing a vector $\mathbf{x}$ within a set of possible movies. Denote as $r(\mathbf{x})$ the rating assigned by a user to movie $\mathbf{x}$. A recommendation is deemed to be unsafe if the user assigns it a rating strictly smaller than 4, i.e., if $r(\mathbf{x}) < 4$. Accordingly, we set both objective function $f(\mathbf{x})$ and constraint function $q(\mathbf{x})$ to be equal to $f(\mathbf{x}) = q(\mathbf{x}) = r(\mathbf{x}) - 4$. We focus on the deterministic safety constraint (6.7), since the ratings are assumed to be observed with no noise.

To define a GP model for the function that maps a movie feature vector $\mathbf{x}$ to a rating $r(\mathbf{x})$, we need to specify a kernel function, which describes the similarity between movies. As in [167], we adopt the linear kernel

$$\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^{\mathsf{T}} \mathbf{x}', \tag{6.45}$$

for any two movie feature vectors $\mathbf{x}$ and $\mathbf{x}'$.

The feature vectors $\mathbf{x}$ for movies are optimized using the MovieLens-100k dataset [57], which includes sparse rating observations of 1,680 movies from 943 users. Specifically, as in [167], we randomly select 200 users to form the training data set, and we set $d = 20$ for the size of the feature vectors. Training applies the standard matrix factorization algorithm [93]. For testing, we pick the 10 test users, not selected for training, that have the most rated movies, and remove the movies with no rating from the possible selections.

Since the true underlying function that maps movie feature vector $\mathbf{x}$ to rating $r(\mathbf{x})$ is unknown, it is not possible to evaluate the RKHS norm $||q||_\kappa$ in (6.16) required by SAFEOPT. Accordingly, as in [13], we set $B = 3$ a priori for SAFEOPT. In this experiment, we run both SAFEOPT and D-SAFE-BOCP for $T = 100$ iterations on the selected 10 test users. We randomly select a movie rated as 4 for each test user as the initial starting point $\mathbf{x}_0$, and set the update rate $\eta = 10$ for D-SAFE-BOCP.

To evaluate the performance of both schemes, we show in Fig. 6.9 the histograms of the ratings across all selected movies during the optimization procedure. The vertical dashed line represents the safety threshold between safe and unsafe recommendations. The marker on the horizontal axis marks the average rating. For D-SAFE-BOCP we have the

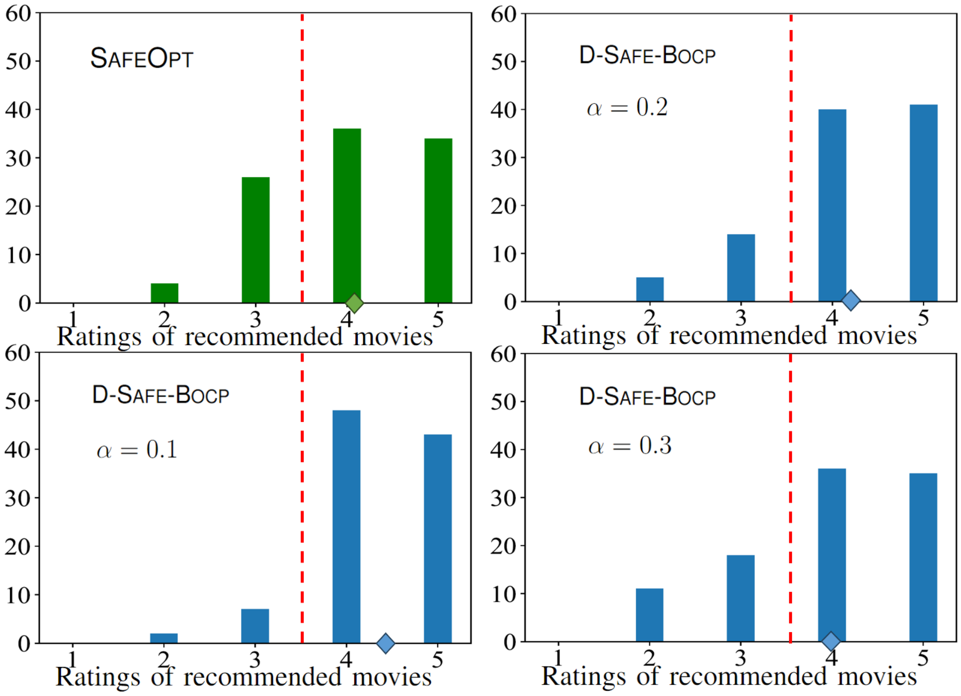

Figure 6.9 Histograms of the ratings of recommended movies by SAFEOPT, as well by D-SAFE-BOCP under different target violation rates α. The dashed lines represent the safety threshold for the recommendations, and the marker on the horizontal axis represents the average rating of the recommendations.

flexibility to vary the target violation rate α, while we recall that for SAFEOPT the target is $\alpha = 0$.

The top-left panel of Fig. 6.9 shows that SAFEOPT does not meet the safety requirement (6.7) with $\alpha = 0$ owing to the mismatch between the assumptions made by the scheme and the true, unknown, constraint function. The remaining panels demonstrate that, in contrast, D-SAFE-BOCP can correctly control the fraction α of unsafe recommendations.

6.9.2 Chemical Reaction Optimization

Finally, we consider the plug flow reactor (PFR) problem introduced in [79], which seeks for optimal chemical reaction parameters $\mathbf{x} \in [140, 200] \times (0, 1] \subset \mathbb{R}^2$, with the first dimension being the temperature ($^\circ C$) and the second being the pH value. The goal is to maximize the yield (%), which we set as the objective $f(\mathbf{x})$, while keeping an acceptable selectivity level (%), which we denote as $s(\mathbf{x})$. We refer to [79] for a precise definition of these terms.

A reaction vector is deemed to be unsafe if the resulting selectivity level is lower than the corresponding yield, hence we define the constraint function as $q(\mathbf{x}) = s(\mathbf{x}) - f(\mathbf{x})$. We assume the presence of non-zero Gaussian observation noise z_t for the constraint function, i.e., $\sigma_q^2 > 0$. Accordingly, we focus on the probabilistic safety constraint (6.8),

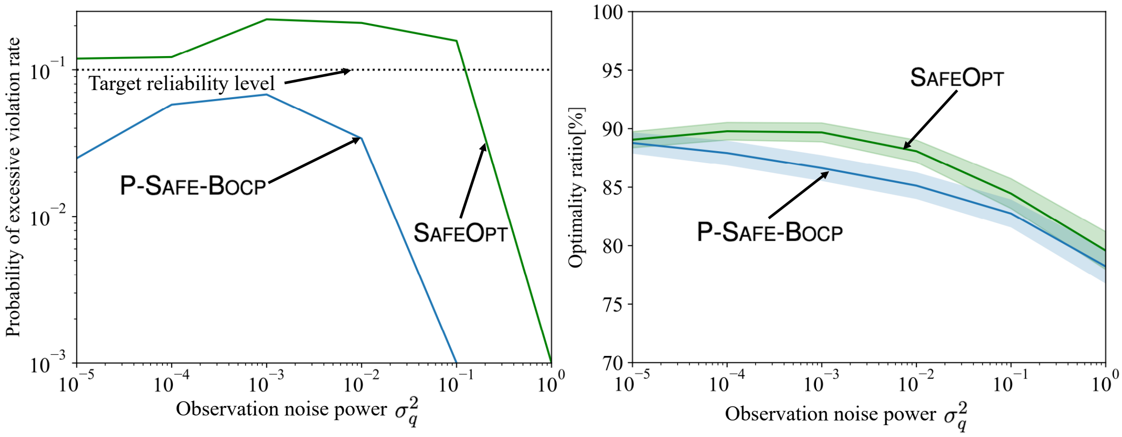

Figure 6.10 Probability of excessive violation rate (6.8) (left) and optimality ratio (6.3) (right) as a function of constraint observation noise power σ_q^2, with update rate $\eta = 2$, RKHS norm bound $B = 3$, and kernel bandwidth $h = 1/2.88$ for the chemical reaction problem.

and compare the performance of SAFEOPT and P-SAFE-BOCP. We adopt GP surrogates model for both $f(\mathbf{x})$ and $q(\mathbf{x})$ with RBF kernel having bandwidth $h = 1/2.88$.

Similar to Sec. 6.9.1, since the smoothness property of the true underlying functions $q(\mathbf{x})$ is unknown, we assume the constant $B = 3$ for SAFEOPT [13]. The initial decision $\mathbf{x}_0$ is randomly chosen among the a priori known safe decisions that satisfy the constraint $q(\mathbf{x_0}) \geq 0$, and we set the total number of optimization round to be $T = 50$. Other settings are as in Sec. 6.8.1.

In a similar manner to Sec. 6.8.3, we demonstrate the excess violation rate probability (6.8) and the optimality ratio in Fig. 6.10 as a function of the observation noise power σ_q^2. Confirming the discussion in Sec. 6.8.3 and the theory, P-SAFE-BOCP is seen to meet the probabilistic safety constraint (6.8) irrespective of observation noise power, while SAFEOPT can only attain an excess violation rate probability below the target $1-\delta$ when the observation noise power is sufficiently large.

6.10 Conclusions

In this chapter, we have introduced SAFE-BOCP, a novel BO-based zero-th order sequential optimizer that provably guarantees safety requirements irrespective of the properties of the constraint function. The key mechanism underlying SAFE-BOCP adapts the level of pessimism adopted during the exploration of the search space on the basis of noisy safety feedback received by the system. From synthetic experiment to real-world applications, we have demonstrated that the proposed SAFE-BOCP performs competitively with state-of-the-art schemes in terms of optimality ratio, while providing for the first time assumption-free safety guarantees.

Although in this work we have built on SAFEOPT for the acquisition process, the proposed framework could be generalized directly to any other Safe-BO schemes, such as GOOSE[179]. Other possible extensions include accounting for multiple constraints, as well as taking into account contextual information during the optimization process [191]. From the application perspective, it would be interesting to investigate the reliability of SAFE-BOCP in more complicated real world systems, such as beamforming design for multi-user URLLC System [58], robust transmission design for IRS-aided secure communications [64], or image target detection [107].

Chapter 7

Conclusions

7.1 Summary of Thesis Achievements

The increasing scalability of machine learning frameworks and real-world engineering system structures fuels the growth on complexity of the black-box optimization problems in these systems. Therefore, the demand for the system designs' reliable performance with generalization on diverse conditions is increasingly important. In this thesis, we investigated a number of possibilities to improve the efficiency, adaptation and reliability of Bayes optimizers.

In Chapter 3, we have studied the fundamentals of GP, the typical surrogate model for BO in later chapters. As any acquisition functions in BO rely on the statistical inference provided by the surrogate model, calibrations over the surrogate model is the most straightforward way to achieve the required data efficiency and adaptation properties for BO. We proposed WFEM, a novel transfer meta-learning approach that generalizes the IMRM via optimizing the free energy objective on the distribution over shared inter-task variables.

In Chapter 4, we considered a real-world wireless communication optimization problem involving radio resource allocation in multi-cell multi-antenna systems. Due to the mobility of the user devices and the inherent randomness in wireless channels, the optimal design for radio resource allocation shifts over time. In order to improve the data efficiency and adaptation of BO in this application, we propose the use of meta-learning to transfer knowledge from data collected from related, but distinct, configurations in order to speed up optimization on new network configurations. Furthermore, we introduce novel contextual meta-optimizers, in which transfer of knowledge across optimization tasks occurs at the level of a mapping from network interference graph based contextual information to resource-allocation design variables. The experiments results provide insights into the potential benefits of meta-learning and contextual optimization strategies.

In Chapter 5, we turned to the scenarios where optimization tasks arrive in a sequence with a fixed evaluation budget and the corresponding objective functions can be approximated into multiple fidelity levels. Information collected at lower fidelity levels can be leveraged to accelerate the optimization process when viewed as a function of the overall cost budget for evaluating the objective function. However, existing strategies only focus on maximizing the information accrued about the optimal candidate solution for the current optimization task. To further enable the efficiency of MFBO, we introduced MFT-MES, a novel information-theoretic acquisition function that balances the need to acquire information about the current task with goal of collecting information transferable to future tasks. We have shown that, by integrating the shared inter-task latent variables into the acquisition function design and updating in Bayesian principles, the performance gain is significant compared to the state-of-art schemes that do not cater to the acquisition of transferable knowledge across tasks.

Finally, in Chapter 6, we investigated the safe optimization problems where the optimizer not only receives objective feedback, but also detects the safety metric of the attempted candidate solution. As the prior works either treat the constraint function as a regularization in the acquisition process without any formal safety guarantee, or make statistical assumptions on the constraint function to achieve theoretical guarantees on all candidate solutions being safe. We proposed SAFE-BOCP, providing for the first time assumptions-free guarantees on the safety level of the attempted candidate solutions, while enabling any non-zero target safety violation level. The experiments demonstrated that the proposed methods perform competitively with state-of-the-art schemes in terms of optimality ratio and safety control.

7.2 Open Research Questions

Many aspects of BO included in this thesis or beyond could be further investigated. In this section, we detail some of these open questions as below.

- As mentioned in Appendix A, federated learning has been widely studied for implementation on wireless communication systems. Recent works on federated Bayesian optimization assume transferable knowledge among different objective functions at local agents, in the form of surrogate model hyperparameters [214], random Fourier features [30], or augmented Lagrangian parameters [89]. An important development would be to study the possibilities of applying federated Bayesian optimization in the next generation wireless communication optimization problems.

- All the optimization problem formulations considered in this thesis allow the optimizer obtain new observation data from the objective function. However, offline or simulation based Bayesian optimization [127] is also important when the optimizer no longer has access to the objective function. An important topic would be how to improve the performance of the optimizer on target objective without obtaining new data.

- This thesis mainly focus on using Gaussian process as the surrogate model, while other choices including Bayesian neural networks [87], gradient boosting machines [49], and tree-structured Parzen estimator [12] could also be utilized to provide statistical inference for Bayesian optimization. Exploring non-GP surrogate models for enhancing efficiency, scalability and reliability of Bayesian optimization would be an important direction in various real-world optimization problems.

Bibliography

[1] Acar, D. A. E., Zhu, R., and Saligrama, V. (2021). Memory efficient online meta learning. In *Proceedings of International Conference on Machine Learning*, pages 32–42.

[2] Alvarez, M. A. and Lawrence, N. D. (2011). Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research*, 12:1459–1500.

[3] Alwasel, K., Jha, D. N., Habeeb, F., Demirbaga, U., Rana, O., Baker, T., Dustdar, S., Villari, M., James, P., Solaiman, E., et al. (2021). IoTSim-Osmosis: A framework for modeling and simulating IoT applications over an edge-cloud continuum. *Journal of Systems Architecture*, 116:101956.

[4] Amiri, M. M. and Gündüz, D. (2020). Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air. *IEEE Transactions on Signal Processing*, 68:2155–2169.

[5] Amit, R. and Meir, R. (2018). Meta-learning by adjusting priors based on extended PAC-Bayes theory. In *Proceedings of International Conference on Machine Learning*, pages 205–214, Stockholm, Sweden.

[6] Anastasiou, A., Barp, A., Briol, F.-X., Ebner, B., Gaunt, R. E., Ghaderinezhad, F., Gorham, J., Gretton, A., Ley, C., Liu, Q., et al. (2023). Stein's method meets computational statistics: A review of some recent developments. *Statistical Science*, 38(1):120–139.

[7] Angelino, E., Johnson, M. J., Adams, R. P., et al. (2016). Patterns of scalable Bayesian inference. *Foundations and Trends® in Machine Learning*, 9(2-3):119–247.

[8] Angelopoulos, A. N. and Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*.

[9] Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyperparameters of Gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66:55–69.

[10] Baxter, J. (2000). A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198.

[11] Bengio, Y., Lahlou, S., Deleu, T., Hu, E. J., Tiwari, M., and Bengio, E. (2023). Gflownet foundations. *Journal of Machine Learning Research*, 24(210):1–55.

[12] Bergstra, J., Yamins, D., and Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of International Conference on Machine Learning*, pages 115–123, Atlanta, USA.

[13] Berkenkamp, F., Krause, A., and Schoellig, A. P. (2023). Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics. *Machine Learning*, 112(10):3713–3747.

[14] Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

[15] Bonilla, E. V., Chai, K., and Williams, C. (2007). Multi-task Gaussian process prediction. In *Proceedings of Advances in Neural Information Processing Systems*, volume 20, Vancouver, Canada.

[16] Boutilier, C., Hsu, C.-W., Kveton, B., Mladenov, M., Szepesvari, C., and Zaheer, M. (2020). Differentiable meta-learning of bandit policies. In *Proceedings of Advances in Neural Information Processing Systems*, volume 33, pages 2122–2134.

[17] Bubeck, S., Munos, R., and Stoltz, G. (2009). Pure exploration in multi-armed bandits problems. In *Proceedings of International Conference on Algorithmic Learning Theory*, pages 23–37, Porto, Portugal.

[18] Cao, X., Zhu, G., Xu, J., Wang, Z., and Cui, S. (2021). Optimized power control design for over-the-air federated edge learning. *IEEE Journal on Selected Areas in Communications*, 40(1):342–358.

[19] Castellanos, C. U., Villa, D. L., Rosa, C., Pedersen, K. I., Calabrese, F. D., Michaelsen, P.-H., and Michel, J. (2008). Performance of uplink fractional power control in UTRAN LTE. In *Proceedings of IEEE Vehicular Technology Conference VTC Spring*, pages 2517–2521, Singapore.

[20] Chen, J., Park, S., and Simeone, O. (2024). Knowing when to stop: Delay-adaptive spiking neural network classifiers with reliability guarantees. *IEEE Journal of Selected Topics in Signal Processing*, pages 1–15.

[21] Chen, J., Skatchkovsky, N., and Simeone, O. (2023a). Neuromorphic wireless cognition: Event-driven semantic communications for remote inference. *IEEE Transactions on Cognitive Communications and Networking*, 9(2):252–265.

[22] Chen, L., Jose, S. T., Nikoloska, I., Park, S., Chen, T., Simeone, O., et al. (2023b). Learning with limited samples: Meta-learning and applications to communication systems. *Foundations and Trends® in Signal Processing*, 17(2):79–208.

[23] Chen, W., Hong, W., Zhang, H., Yang, P., and Tang, K. (2022). Multi-fidelity simulation modeling for discrete event simulation: An optimization perspective. *IEEE Transactions on Automation Science and Engineering*, 20(2):1156–1169.

[24] Chen, X., Wu, S. Z., and Hong, M. (2020). Understanding gradient clipping in private SGD: A geometric perspective. In *Proceedings of Advances in Neural Information Processing Systems*, volume 33, pages 13773–13782.

[25] Collins, L., Mokhtari, A., and Shakkottai, S. (2020). Task-robust model-agnostic meta-learning. In *Proceedings of Advances in Neural Information Processing Systems*, volume 33, pages 18860–18871.

[26] Cortes, C. L., Lefebvre, P., Lauk, N., Davis, M. J., Sinclair, N., Gray, S. K., and Oblak, D. (2022). Sample-efficient adaptive calibration of quantum networks using Bayesian optimization. *Physical Review Applied*, 17(3):034067.

[27] Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory*. (Wiley Series in Telecommunications and Signal Processing), 2nd ed. New York, NY, USA: Wiley.

[28] Crowl, D. A. and Louvar, J. F. (2001). *Chemical process safety: fundamentals with applications*. Pearson Education.

[29] Cui, W., Shen, K., and Yu, W. (2020). Deep learning for robust power control for wireless networks. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8554–8558.

[30] Dai, Z., Low, B. K. H., and Jaillet, P. (2021). Differentially private federated Bayesian optimization with distributed exploration. In *Proceedings of Advances in Neural Information Processing Systems*, volume 34, pages 9125–9139.

[31] Damianou, A. and Lawrence, N. D. (2013). Deep Gaussian processes. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pages 207–215, Arizona, USA.

[32] Demetrio, L., Biggio, B., Lagorio, G., Roli, F., and Armando, A. (2021). Functionality-preserving black-box optimization of adversarial Windows malware. *IEEE Transactions on Information Forensics and Security*, 16:3469–3478.

[33] Ding, Y., Kim, M., Kuindersma, S., and Walsh, C. J. (2018). Human-in-the-loop optimization of hip assistance with a soft exosuit during walking. *Science Robotics*, 3(15):eaar5438.

[34] Dixon, L. C. W. (1978). The global optimization problem: an introduction. *Towards Global Optimiation 2*, pages 1–15.

[35] Dusenberry, M., Jerfel, G., Wen, Y., Ma, Y., Snoek, J., Heller, K., Lakshminarayanan, B., and Tran, D. (2020). Efficient and scalable Bayesian neural nets with rank-1 factors. In *Proceedings of International Conference on Machine Learning*, pages 2782–2792.

[36] Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

[37] Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. (2019). Scalable global optimization via local Bayesian optimization. *Advances in Neural Information Processing Systems*, 32, Vancouver, Canada.

[38] Fannjiang, C., Bates, S., Angelopoulos, A. N., Listgarten, J., and Jordan, M. I. (2022). Conformal prediction under feedback covariate shift for biomolecular design. *Proceedings of the National Academy of Sciences*, 119(43):e2204569119.

[39] Feldman, S., Einbinder, B.-S., Bates, S., Angelopoulos, A. N., Gendler, A., and Romano, Y. (Limassol, Cyprus, 2023). Conformal prediction is robust to dispersive label noise. In *Proceedings of Symposium on Conformal and Probabilistic Prediction with Applications*, pages 624–626.

[40] Feldman, S., Ringel, L., Bates, S., and Romano, Y. (2022). Achieving risk control in online learning settings. *arXiv preprint arXiv:2205.09095*.

[41] Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of International Conference on Machine Learning*, pages 1126–1135, Sydney, Australia.

[42] Finn, C., Rajeswaran, A., Kakade, S., and Levine, S. (2019). Online meta-learning. In *Proceedings of International Conference on Machine Learning*, pages 1920–1930, California.

[43] Fishman, G. (2013). *Monte Carlo: concepts, algorithms, and applications*. Springer Science & Business Media.

[44] Forrester, A. I., Sóbester, A., and Keane, A. J. (2007). Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3251–3269.

[45] Fortuin, V., Strathmann, H., and Rätsch, G. (2019). Meta-learning mean functions for Gaussian processes. *arXiv preprint arXiv:1901.08098*.

[46] Frazier, P., Powell, W., and Dayanik, S. (2009). The knowledge-gradient policy for correlated normal beliefs. *INFORMS journal on Computing*, 21(4):599–613.

[47] Frazier, P. I. (2018). A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.

[48] Frazier, P. I., Powell, W. B., and Dayanik, S. (2008). A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439.

[49] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

[50] Gai, Y., Krishnamachari, B., and Jain, R. (2012). Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5):1466–1478.

[51] Gandikota, V., Kane, D., Maity, R. K., and Mazumdar, A. (2022). vqSGD: Vector quantized stochastic gradient descent. *IEEE Transactions on Information Theory*, 68(7):4573–4587.

[52] Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. (2018). Gpytorch: Blackbox matrix-matrix Gaussian process inference with gpu acceleration. In *Proceedings of Advances in Neural Information Processing Systems*, volume 31, Montreal, Canada.

[53] Gardner, J. R., Kusner, M. J., Xu, Z., Weinberger, K. Q., and Cunningham, J. P. (2014). Bayesian optimization with inequality constraints. In *Proceedings of International Conference on Machine Learning*, pages 937–945, Beijing, China.

[54] Gelbart, M. A., Snoek, J., and Adams, R. P. (2014). Bayesian optimization with unknown constraints. In *Proceedings of Conference on Uncertainty in Artificial Intelligence*, pages 250–259, Quebec, Canada.

[55] Gibbs, I. and Candes, E. (2021). Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672.

[56] Gündüz, D., de Kerret, P., Sidiropoulos, N. D., Gesbert, D., Murthy, C. R., and van der Schaar, M. (2019). Machine learning in the air. *IEEE Journal on Selected Areas in Communications*, 37(10):2184–2199.

[57] Harper, F. and Konstan, J. (2016). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TIIS)*, 5(4).

[58] He, S., Xiong, S., An, Z., Zhang, W., Huang, Y., and Zhang, Y. (2022). An unsupervised deep unrolling framework for constrained optimization problems in wireless networks. *IEEE Transactions on Wireless Communications*, 21(10):8552–8564.

[59] He, S., Xiong, S., Ou, Y., Zhang, J., Wang, J., Huang, Y., and Zhang, Y. (2021). An overview on the application of graph neural networks in wireless networks. *IEEE Open Journal of the Communications Society*, 2:2547–2565.

[60] Hennig, P. and Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6).

[61] Hernández-Lobato, J. M., Gelbart, M., Hoffman, M., Adams, R., and Ghahramani, Z. (2015). Predictive entropy search for Bayesian optimization with unknown constraints. In *Proceedings of International Conference on Machine Learning*, pages 1699–1707.

[62] Hernández-Lobato, J. M., Gelbart, M. A., Adams, R. P., Hoffman, M. W., and Ghahramani, Z. (2016). A general framework for constrained Bayesian optimization using information-based search. *Journal of Machine Learning Research*, 17(1):5549–5601.

[63] Highfield, R. and Coveney, P. (2023). Virtual you: how building your digital twin will revolutionize medicine and change your life. *Princeton University Press.*

[64] Hong, S., Pan, C., Zhou, G., Ren, H., and Wang, K. (2024). Outage constrained robust transmission design for IRS-aided secure communications with direct communication links. *IEEE Transactions on Communications*, 72(3):1548–1564.

[65] Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. (2021). Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169.

[66] Hou, Y., Wang, Y., Xiang, T., Xie, J., Zhao, J., He, W., Huang, W., and Wu, D. (2024). An efficient deconvolution method for automatic detection of bearing localized defect based on Bayesian optimization. *IEEE Transactions on Instrumentation and Measurement*, 73:1–15.

[67] Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745.*

[68] Hoydis, J., Aoudia, F. A., Cammerer, S., Nimier-David, M., Binder, N., Marcus, G., and Keller, A. (2023). Sionna rt: Differentiable ray tracing for radio propagation modeling. *arXiv preprint arXiv:2303.11103.*

[69] Hsieh, B.-J., Hsieh, P.-C., and Liu, X. (2021). Reinforced few-shot acquisition function learning for Bayesian optimization. In *Proceedings of Advances in Neural Information Processing Systems*, volume 34, pages 7718–7731.

[70] Hu, J., Liu, G., Ma, Z., Xiao, M., and Fan, P. (2023). Low-complexity resource allocation for uplink RSMA in future 6G wireless networks. *IEEE Wireless Communications Letters*, 13(2):565–569.

[71] Huang, D., Allen, T. T., Notz, W. I., and Miller, R. A. (2006). Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32:369–382.

[72] Jin, R., Huang, Y., He, X., Dai, H., and Wu, T. (2020). Stochastic-sign SGD for federated learning with theoretical guarantees. *arXiv preprint arXiv:2002.10940.*

[73] Johnston, J., Liu, X.-Y., Wu, S., and Wang, X. (2024). A curriculum learning approach to optimization with application to downlink beamforming. *IEEE Transactions on Signal Processing*, 72:84–98.

[74] Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492.

[75] Jose, S. T. and Simeone, O. (2021a). Free energy minimization: A unified framework for modeling, inference, learning, and optimization [lecture notes]. *IEEE Signal Processing Magazine*, 38(2):120–125.

[76] Jose, S. T. and Simeone, O. (2021b). Information-theoretic bounds on transfer generalization gap based on Jensen-Shannon divergence. In *Proceedings of European Signal Processing Conference (EUSIPCO)*, pages 1461–1465, Dublin, Ireland.

[77] Jose, S. T., Simeone, O., and Durisi, G. (2021). Transfer meta-learning: Information-theoretic bounds and information meta-risk minimization. *IEEE Transactions on Information Theory*, 68(1):474–501.

[78] Kandasamy, K., Dasarathy, G., Schneider, J., and Póczos, B. (2017). Multi-fidelity Bayesian optimisation with continuous approximations. In *Proceedings of International Conference on Machine Learning*, pages 1799–1808, Sydney, Australia.

[79] Kang, P.-L., Shang, C., and Liu, Z.-P. (2019). Glucose to 5-hydroxymethylfurfural: origin of site-selectivity resolved by machine learning based reaction sampling. *Journal of the American Chemical Society*, 141(51):20525–20536.

[80] Karvonen, T. (2021). Estimation of the scale parameter for a misspecified Gaussian process model. *arXiv preprint arXiv:2110.02810*.

[81] Kato, N., Mao, B., Tang, F., Kawamoto, Y., and Liu, J. (2020). Ten challenges in advancing machine learning technologies toward 6G. *IEEE Wireless Communications*, 27(3):96–103.

[82] Kennedy, M. C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13.

[83] Khan, M. E. and Rue, H. (2023). The Bayesian learning rule. *Journal of Machine Learning Research*, 24(281):1–46.

[84] Knoblauch, J., Jewson, J., and Damoulas, T. (2022). An optimization-centric view on Bayes' rule: Reviewing and generalizing variational inference. *Journal of Machine Learning Research*, 23(132):1–109.

[85] Koda, Y., Yamamoto, K., Nishio, T., and Morikura, M. (2020). Differentially private aircomp federated learning with power adaptation harnessing receiver noise. In *proceedings of IEEE Global Communications Conference (GLOBECOM)*, pages 1–6.

[86] König, C., Ozols, M., Makarova, A., Balta, E. C., Krause, A., and Rupenyan, A. (2023). Safe risk-averse Bayesian optimization for controller tuning. *IEEE Robotics and Automation Letters*.

[87] Kononenko, I. (1989). Bayesian neural networks. *Biological Cybernetics*, 61(5):361–370.

[88] Krause, A. and Ong, C. (2011). Contextual Gaussian process bandit optimization. In *Proceedings of Advances in Neural Information Processing Systems*, volume 24, Granada, Spain.

[89] Krishnamoorthy, D. and Paulson, J. A. (2023). Multi-agent black-box optimization using a Bayesian approach to alternating direction method of multipliers. In *Proceedings of International Federation of Automatic Control*, volume 56, pages 2232–2237, Yokohama, Japan.

[90] Kumar, A. and Kumar, K. (2022). Deep learning-based joint NOMA signal detection and power allocation in cognitive radio networks. *IEEE Transactions on Cognitive Communications and Networking*, 8(4):1743–1752.

[91] Lam, R., Allaire, D. L., and Willcox, K. E. (2015). Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *Proceedings of 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pages 0143, Florida, USA.

[92] Le Gratiet, L. and Garnier, J. (2014). Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification*.

[93] Lee, D. and Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *Proceedings of Advances in Neural Information Processing Systems*, volume 13, Denver, USA.

[94] Lee, H. B., Lee, H., Na, D., Kim, S., Park, M., Yang, E., and Hwang, S. J. (2020). Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. In *Proceedings of International Conference on Learning Representations*.

[95] Lee, W., Kim, M., and Cho, D.-H. (2018). Transmit power control using deep neural network for underlay device-to-device communication. *IEEE Wireless Communications Letters*, 8(1):141–144.

[96] Lei, B., Kirk, T. Q., Bhattacharya, A., Pati, D., Qian, X., Arroyave, R., and Mallick, B. K. (2021). Bayesian optimization with adaptive surrogate models for automated experimental design. *Npj Computational Materials*, 7(1):194.

[97] Li, H., Jin, Y., and Chai, T. (2024). Evolutionary multi-objective Bayesian optimization based on multisource online transfer learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(1):488–502.

[98] Li, H. and Zhang, J. (2017). Fast source term estimation using the PGA-NM hybrid method. *Engineering Applications of Artificial Intelligence*, 62:68–79.

[99] Liang, F., Shen, C., Yu, W., and Wu, F. (2019). Towards optimal power control via ensembling deep neural networks. *IEEE Transactions on Communications*, 68(3):1760–1776.

[100] Liang, Z., Zhu, Y., Wang, X., Li, Z., and Zhu, Z. (2023). Evolutionary multitasking for multi-objective optimization based on generative strategies. *IEEE Transactions on Evolutionary Computation*, 27(4):1042–1056.

[101] Liu, D. and Simeone, O. (2020). Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control. *IEEE Journal on Selected Areas in Communications*, 39(1):170–185.

[102] Liu, D. and Simeone, O. (2022a). Wireless federated Langevin Monte Carlo: Repurposing channel noise for bayesian sampling and privacy. *IEEE Transactions on Wireless Communications*, 22(5):2946–2961.

[103] Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528.

[104] Liu, Q. and Wang, D. (2016). Stein variational gradient descent: A general purpose Bayesian inference algorithm. *Proceedings of Advances in Neural Information Processing Systems*, 29, Barcelona, Spain.

[105] Liu, W., Zang, X., Li, Y., and Vucetic, B. (2020). Over-the-air computation systems: Optimization, analysis and scaling laws. *IEEE Transactions on Wireless Communications*, 19(8):5488–5502.

[106] Liu, Y. and Simeone, O. (2022b). Learning how to transfer from uplink to downlink via hyper-recurrent neural network for FDD massive MIMO. *IEEE Transactions on Wireless Communications*, 21(10):7975–7989.

[107] Liufu, Y., Jin, L., Xu, J., Xiao, X., and Fu, D. (2021). Reformative noise-immune neural network for equality-constrained optimization applied to image target detection. *IEEE Transactions on Emerging Topics in Computing*, 10(2):973–984.

[108] Lyddon, S. P., Holmes, C., and Walker, S. (2019). General Bayesian updating and the loss-likelihood bootstrap. *Biometrika*, 106(2):465–478.

[109] MacKay, D. J. (1992). Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604.

[110] MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge University Press.

[111] Maddox, W. J., Balandat, M., Wilson, A. G., and Bakshy, E. (2021). Bayesian optimization with high-dimensional outputs. In *Proceedings of Advances in Neural Information Processing Systems*, volume 34, pages 19274–19287.

[112] Maggi, L., Valcarce, A., and Hoydis, J. (2021). Bayesian optimization for radio resource management: Open loop power control. *IEEE Journal on Selected Areas in Communications*, 39(7):1858–1871.

[113] Malkomes, G., Schaff, C., and Garnett, R. (2016). Bayesian optimization for automated model selection. In *Proceedings of Advances in Neural Information Processing Systems*, volume 29, Barcelona, Spain.

[114] Marco, A., von Rohr, A., Baumann, D., Hernández-Lobato, J. M., and Trimpe, S. (2020). Excursion search for constrained Bayesian optimization under a limited budget of failures. *arXiv preprint arXiv:2005.07443*.

[115] Marini, R., Park, S., Simeone, O., and Buratti, C. (2023). Continual meta-reinforcement learning for UAV-aided vehicular wireless networks. In *Proceedings of IEEE International Conference on Communications*, pages 5664–5669, Rome, Italy.

[116] Massart, P. (1990). The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. *The Annals of Probability*, pages 1269–1283.

[117] Matsubara, T., Knoblauch, J., Briol, F.-X., and Oates, C. J. (2023). Generalized Bayesian inference for discrete intractable likelihood. *Journal of the American Statistical Association*, pages 1–11.

[118] Michael, N. E., Hasan, S., Al-Durra, A., and Mishra, M. (2022). Short-term solar irradiance forecasting based on a novel Bayesian optimized deep long short-term memory neural network. *Applied Energy*, 324:119727.

[119] Michaud, R. O. and Michaud, R. O. (2008). *Efficient asset management: a practical guide to stock portfolio optimization and asset allocation*. Oxford University Press.

[120] Mikkola, P., Martinelli, J., Filstroff, L., and Kaski, S. (2023). Multi-fidelity Bayesian optimization with unreliable information sources. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pages 7425–7454, Valencia, Spain.

[121] Močkus, J. (1975). On Bayesian methods for seeking the extremum. In *Proceedings of Optimization Techniques IFIP Technical Conference: Novosibirsk, Russia*, pages 400–404.

[122] Mockus, J. (1989). Global optimization and the Bayesian approach. *Bayesian Approach to Global Optimization: Theory and Applications*, pages 1–3.

[123] Moshksar, K. and Khandani, A. K. (2016). Arbitrarily tight bounds on differential entropy of Gaussian mixtures. *IEEE Transactions on Information Theory*, 62(6):3340–3354.

[124] Moss, H. B., Aggarwal, V., Prateek, N., González, J., and Barra-Chicote, R. (2020). Boffin tts: Few-shot speaker adaptation by Bayesian optimization. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7639–7643.

[125] Moss, H. B., Leslie, D. S., Gonzalez, J., and Rayson, P. (2021a). Gibbon: General-purpose information-based Bayesian optimisation. *Journal of Machine Learning Research*, 22(1):10616–10664.

[126] Moss, H. B., Leslie, D. S., and Rayson, P. (2021b). Mumbo: Multi-task max-value Bayesian optimization. In *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 447–462, Ghent, Belgium.

[127] Naveiro, R. and Tang, B. (2024). Simulation based Bayesian optimization. *arXiv preprint arXiv:2401.10811*.

[128] Nazer, B. and Gastpar, M. (2007). Computation over multiple-access channels. *IEEE Transactions on Information Theory*, 53(10):3498–3516.

[129] Nielsen, F. and Sun, K. (2016). Guaranteed bounds on information-theoretic measures of univariate mixtures using piecewise log-sum-exp inequalities. *Entropy*, 18(12):442.

[130] Nikoloska, I. and Simeone, O. (2022a). Bayesian active meta-learning for black-box optimization. In *Proceedings of IEEE International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, pages 1–5, Oulu, Finland.

[131] Nikoloska, I. and Simeone, O. (2022b). Modular meta-learning for power control via random edge graph neural networks. *IEEE Transactions on Wireless Communications*, 22(1):457–470.

[132] Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

[133] Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71.

[134] Park, J., Samarakoon, S., Bennis, M., and Debbah, M. (2019). Wireless network intelligence at the edge. *Proceedings of the IEEE*, 107(11):2204–2239.

[135] Park, S., Jang, H., Simeone, O., and Kang, J. (2020a). Learning to demodulate from few pilots via offline and online meta-learning. *IEEE Transactions on Signal Processing*, 69:226–239.

[136] Park, S. and Simeone, O. (2022). Predicting flat-fading channels via meta-learned closed-form linear filters and equilibrium propagation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8817–8821, Singapore.

[137] Park, S., Simeone, O., and Kang, J. (2020b). Meta-learning to communicate: Fast end-to-end training for fading channels. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5075–5079, Barcelona, Spain.

[138] Pedersen, J. K., Clausen, C. M., Krysiak, O. A., Xiao, B., Batchelor, T. A., Löffler, T., Mints, V. A., Banko, L., Arenz, M., Savan, A., et al. (2021). Bayesian optimization of high-entropy alloy compositions for electrocatalytic oxygen reduction. *Angewandte Chemie*, 133(45):24346–24354.

[139] Pentina, A. and Lampert, C. (2014). A PAC-Bayesian bound for lifelong learning. In *Proceedings of International Conference on Machine Learning*, pages 991–999, Beijing, China.

[140] Polese, M., Bonati, L., D'oro, S., Basagni, S., and Melodia, T. (2023). Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges. *IEEE Communications Surveys & Tutorials*, 25(2):1376–1411.

[141] Poloczek, M., Wang, J., and Frazier, P. (2017). Multi-information source optimization. In *Proceedings of Advances in Neural Information Processing Systems*, volume 30, California, USA.

[142] Rasmussen, C. E. (2004). *Gaussian Processes in Machine Learning*, pages 63–71. Springer.

[143] Riis, C., Antunes, F., Hüttel, F., Lima Azevedo, C., and Pereira, F. (2022). Bayesian active learning with fully Bayesian Gaussian processes. In *Proceedings of Advances in Neural Information Processing Systems*, volume 35, pages 12141–12153.

[144] Rothfuss, J., Fortuin, V., Josifoski, M., and Krause, A. (2021a). Pacoh: Bayes-optimal meta-learning with PAC-guarantees. In *Proceedings of International Conference on Machine Learning*, pages 9116–9126.

[145] Rothfuss, J., Heyn, D., Krause, A., et al. (2021b). Meta-learning reliable priors in the function space. In *Proceedings of Advances in Neural Information Processing Systems*, volume 34, pages 280–293.

[146] Rothfuss, J., Koenig, C., Rupenyan, A., and Krause, A. (2023). Meta-learning priors for safe Bayesian optimization. In *Proceedings of Conference on Robot Learning*, pages 237–265, Georgia, USA.

[147] Ruah, C., Simeone, O., and Al-Hashimi, B. (2023a). A Bayesian framework for digital twin-based control, monitoring, and data collection in wireless systems. *IEEE Journal on Selected Areas in Communications*, 41(10):3146–3160.

[148] Ruah, C., Simeone, O., Hoydis, J., and Al-Hashimi, B. (2023b). Calibrating wireless ray tracing for digital twinning using local phase error estimates. *arXiv preprint arXiv:2312.12625.*

[149] Schmidhuber, J. (1987). *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook.* PhD thesis, Technische Universität München.

[150] Seif, M., Tandon, R., and Li, M. (2020). Wireless federated learning with local differential privacy. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 2604–2609.

[151] Sery, T., Shlezinger, N., Cohen, K., and Eldar, Y. C. (2021). Over-the-air federated learning from heterogeneous data. *IEEE Transactions on Signal Processing*, 69:3796–3811.

[152] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

[153] Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., and Borgwardt, K. (2009). Efficient graphlet kernels for large graph comparison. In *Proceeedings of Artificial Intelligence and Statistics*, pages 488–495, Florida, USA.

[154] Shields, B. J., Stevens, J., Li, J., Parasram, M., Damani, F., Alvarado, J. I. M., Janey, J. M., Adams, R. P., and Doyle, A. G. (2021). Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96.

[155] Simeone, O. (2018). A very brief introduction to machine learning with applications to communication systems. *IEEE Transactions on Cognitive Communications and Networking*, 4(4):648–664.

[156] Simeone, O. (2022). *Machine Learning for Engineers.* Cambridge University Press.

[157] Simeone, O., Park, S., and Kang, J. (2020). From learning to meta-learning: Reduced training overhead and complexity for communication systems. In *Proceedings of IEEE 6G Wireless Summit (6G SUMMIT)*, pages 1–5.

[158] Slivkins, A. et al. (2019). Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286.

[159] Sloman, S. J., Bharti, A., and Kaski, S. (2023). The fundamental dilemma of Bayesian active meta-learning. *arXiv preprint arXiv:2310.14968.*

[160] Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., and Adams, R. (2015). Scalable Bayesian optimization using deep neural networks. In *Proceedings of International Conference on Machine Learning*, pages 2171–2180, Lille, France.

[161] Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of International Conference on Machine Learning*, pages 1015–1022, Haifa, Israel.

[162] Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. W. (2012). Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265.

[163] Stanton, S., Maddox, W., and Wilson, A. G. (2023). Bayesian optimization with conformal prediction sets. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pages 959–986, Valencia, Spain.

[164] Staudenmaier, N., Vijayakumar-Sreeja, A., Genov, G., Cohen, D., Findler, C., Lang, J., Retzker, A., Jelezko, F., and Oviedo-Casado, S. (2023). Optimal sensing protocol for statistically polarized nano-NMR with NV centers. *Physical Review Letters*, 131(15):150801.

[165] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2):111–133.

[166] Sugiyama, M., Krauledat, M., and Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5).

[167] Sui, Y., Gotovos, A., Burdick, J., and Krause, A. (2015). Safe exploration for optimization with Gaussian processes. In *Proceedings of International Conference on Machine Learning*, pages 997–1005, Lille, France.

[168] Sui, Y., Zhuang, V., Burdick, J., and Yue, Y. (2018). Stagewise safe Bayesian optimization with Gaussian processes. In *Proceedings of International Conference on Machine Learning*, pages 4781–4789, Stockholm, Sweden.

[169] Sun, H., Chen, X., Shi, Q., Hong, M., Fu, X., and Sidiropoulos, N. D. (2018). Learning to optimize: Training deep neural networks for interference management. *IEEE Transactions on Signal Processing*, 66(20):5438–5453.

[170] Sun, Y., Peng, M., Zhou, Y., Huang, Y., and Mao, S. (2019). Application of machine learning in wireless networks: Key techniques and open issues. *IEEE Communications Surveys & Tutorials*, 21(4):3072–3108.

[171] Swanson, K., Liu, G., Catacutan, D. B., Arnold, A., Zou, J., and Stokes, J. M. (2024). Generative AI for designing and validating easily synthesizable and structurally novel antibiotics. *Nature Machine Intelligence*, 6:338–353.

[172] Swersky, K., Snoek, J., and Adams, R. P. (2013). Multi-task Bayesian optimization. In *Proceedings of Advances in Neural Information Processing Systems*, volume 26, Nevada, USA.

[173] Syring, N. and Martin, R. (2019). Calibrating general posterior credible regions. *Biometrika*, 106(2):479–486.

[174] Takeno, S., Fukuoka, H., Tsukada, Y., Koyama, T., Shiga, M., Takeuchi, I., and Karasuyama, M. (2020). Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In *Proceedings of International Conference on Machine Learning*, pages 9334–9345.

[175] Tambovskiy, S. S., Fodor, G., and Tullberg, H. (2022). Cell-free data power control via scalable multi-objective Bayesian optimisation. In *Proceedings of IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–6.

[176] Tan, J., Liang, Y.-C., Zhang, L., and Feng, G. (2020). Deep reinforcement learning for joint channel selection and power control in D2D networks. *IEEE Transactions on Wireless Communications*, 20(2):1363–1378.

[177] Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294.

[178] Tse, D. and Viswanath, P. (2005). *Fundamentals of wireless communication*. Cambridge University Press.

[179] Turchetta, M., Berkenkamp, F., and Krause, A. (2019). Safe exploration for interactive machine learning. In *Proceedings of Advances in Neural Information Processing Systems*, volume 32, Vancouver, Canada.

[180] Vilalta, R. and Drissi, Y. (2002). A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95.

[181] Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. In *Proceedings of Advances in Neural Information Processing Systems*, volume 29, Barcelona, Spain.

[182] Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic learning in a random world*, volume 29. Springer.

[183] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011). The caltech-ucsd birds-200-2011 dataset.

[184] Wang, L., Zhang, X., Su, H., and Zhu, J. (2024). A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20.

[185] Wang, Y., Fang, J., Cheng, Y., She, H., Guo, Y., and Zheng, G. (2023). Cooperative end-edge-cloud computing and resource allocation for digital twin enabled 6G industrial iot. *IEEE Journal of Selected Topics in Signal Processing*, pages 1–14.

[186] Wang, Y., Li, R., Dong, H., Ma, Y., Yang, J., Zhang, F., Zhu, J., and Li, S. (2019). Capacity planning and optimization of business park-level integrated energy system based on investment constraints. *Energy*, 189:116345.

[187] Wang, Y.-X., Fienberg, S., and Smola, A. (2015). Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In *Proceedings of International Conference on Machine Learning*, pages 2493–2502, Lille, France.

[188] Wang, Z. and Jegelka, S. (2017). Max-value entropy search for efficient Bayesian optimization. In *Proceedings of International Conference on Machine Learning*, pages 3627–3635, Sydney, Australia.

[189] Wang, Z., Tan, V. Y., and Scarlett, J. (2022). Tight regret bounds for noisy optimization of a Brownian motion. *IEEE Transactions on Signal Processing*, 70:1072–1087.

[190] Weiss, N. A., Holmes, P. T., and Hardy, M. (2006). *A course in probability*. Pearson Addison Wesley Boston, MA, USA:.

[191] Widmer, D., Kang, D., Sukhija, B., Hübotter, J., Krause, A., and Coros, S. (2023). Tuning legged locomotion controllers via safe Bayesian optimization. In *Proceedings of Conference on Robot Learning*, pages 2444–2464, Atlanta, USA.

[192] Wu, J., Toscano-Palmerin, S., Frazier, P. I., and Wilson, A. G. (2020). Practical multi-fidelity Bayesian optimization for hyperparameter tuning. In *Proceedings of Uncertainty in Artificial Intelligence*, pages 788–798.

[193] Wynne, G., Briol, F.-X., and Girolami, M. (2021). Convergence guarantees for Gaussian process means with misspecified likelihoods and smoothness. *Journal of Machine Learning Research*, 22(123):1–40.

[194] Xu, S., Li, J., Cai, P., Liu, X., Liu, B., and Wang, X. (2021). Self-improving photosensitizer discovery system via Bayesian search with first-principle simulations. *Journal of the American Chemical Society*, 143(47):19769–19777.

[195] Yan, J., Lu, Q., and Giannakis, G. B. (2024). Bayesian optimization for online management in dynamic mobile edge computing. *IEEE Transactions on Wireless Communications*, 23(4):3425–3436.

[196] Yanardag, P. and Vishwanathan, S. (2015). Deep graph kernels. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1365–1374, Sydney, Australia.

[197] Yang, H. H., Chen, Z., Quek, T. Q., and Poor, H. V. (2021). Revisiting analog over-the-air machine learning: The blessing and curse of interference. *IEEE Journal of Selected Topics in Signal Processing*, 16(3):406–419.

[198] Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y., and Ahn, S. (2018). Bayesian model-agnostic meta-learning. In *Proceedings of Advances in Neural Information Processing Systems*, volume 31, Montreal Canada.

[199] Yu, C., Cao, J., and Rosendo, A. (2022). Learning to climb: Constrained contextual Bayesian optimisation on a multi-modal legged robot. *IEEE Robotics and Automation Letters*, 7(4):9881–9888.

[200] Yuan, Y., Zheng, G., Wong, K.-K., Ottersten, B., and Luo, Z.-Q. (2020). Transfer learning and meta learning-based fast downlink beamforming adaptation. *IEEE Transactions on Wireless Communications*, 20(3):1742–1755.

[201] Zarini, H., Mili, M. R., Rasti, M., Andreev, S., Nardelli, P. H., and Bennis, M. (2023). Intelligent analog beam selection and beamspace channel tracking in THz massive MIMO with lens antenna array. *IEEE Transactions on Cognitive Communications and Networking*, 9(3):629–646.

[202] Zellner, A. (1988). Optimal information processing and Bayes's theorem. *The American Statistician*, 42(4):278–280.

[203] Zhang, M., Li, H., Pan, S., Lyu, J., Ling, S., and Su, S. (2021a). Convolutional neural networks-based lung nodule classification: A surrogate-assisted evolutionary algorithm for hyperparameter optimization. *IEEE Transactions on Evolutionary Computation*, 25(5):869–882.

[204] Zhang, T. (2006). Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321.

[205] Zhang, W., Derakhshani, M., Zheng, G., Chen, C. S., and Lambotharan, S. (2022). Bayesian optimization of queuing-based multi-channel urllc scheduling. *IEEE Transactions on Wireless Communications.*

[206] Zhang, Y., Jordon, J., Alaa, A. M., and van der Schaar, M. (2019). Lifelong Bayesian optimization. *arXiv preprint arXiv:1905.12280.*

[207] Zhang, Y., Park, S., and Simeone, O. (2024). Bayesian optimization with formal safety guarantees via online conformal prediction. *IEEE Journal of Selected Topics in Signal Processing*, pages 1–15.

[208] Zhang, Y., Simeone, O., Jose, S. T., Maggi, L., and Valcarce, A. (2023). Bayesian and multi-armed contextual meta-optimization for efficient wireless radio resource management. *IEEE Transactions on Cognitive Communications and Networking*, 9(5):1282–1295.

[209] Zhang, Y., Zhang, J., Jin, Y., Buzzi, S., and Ai, B. (2021b). Deep learning-based power control for uplink cell-free massive MIMO systems. In *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, pages 1–6.

[210] Zhou, J., Yang, Z., Si, Y., Kang, L., Li, H., Wang, M., and Zhang, Z. (2020). A trust-region parallel Bayesian optimization method for simulation-driven antenna design. *IEEE Transactions on Antennas and Propagation*, 69(7):3966–3981.

[211] Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., and Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8):1738–1762.

[212] Zhu, G., Du, Y., Gündüz, D., and Huang, K. (2020a). One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis. *IEEE Transactions on Wireless Communications*, 20(3):2120–2135.

[213] Zhu, G., Liu, D., Du, Y., You, C., Zhang, J., and Huang, K. (2020b). Toward an intelligent edge: Wireless communication meets machine learning. *IEEE communications magazine*, 58(1):19–25.

[214] Zhu, H., Wang, X., and Jin, Y. (2023). Federated many-task Bayesian optimization. *IEEE Transactions on Evolutionary Computation*, pages 1–1.

Appendix A

Leveraging Channel Noise for Sampling and Privacy via Quantized Federated Langevin Monte Carlo

A.1 Overview

For engineering applications of artificial intelligence, Bayesian learning holds significant advantages over standard frequentist learning, including the capacity to quantify uncertainty. Langevin Monte Carlo (LMC) is an efficient gradient-based approximate Bayesian learning strategy that aims at producing samples drawn from the posterior distribution of the model parameters. Prior work focused on a distributed implementation of LMC over a multi-access wireless channel via analog modulation. In contrast, this chapter proposes quantized federated LMC (FLMC), which integrates one-bit stochastic quantization of the local gradients with channel-driven sampling. Channel-driven sampling leverages channel noise for the purpose of contributing to Monte Carlo sampling, while also serving the role of privacy mechanism. Analog and digital implementations of wireless LMC are compared as a function of differential privacy (DP) requirements, revealing the advantages of the latter at sufficiently high signal-to-noise ratio.

A.2 Introduction

Federated learning (FL) is a distributed learning paradigm whereby multiple devices coordinate to train a target global model, while avoiding the direct sharing of local data with the cloud [134, 211, 213]. Prior work on wireless FL mainly focuses on conventional *frequentist* learning, which produces point estimates of model parameter vectors by minimizing empirical loss metrics [151, 197, 4, 101, 18, 212]. In many engineering applications characterized by the availability of limited data and by the need to quantify uncertainty,

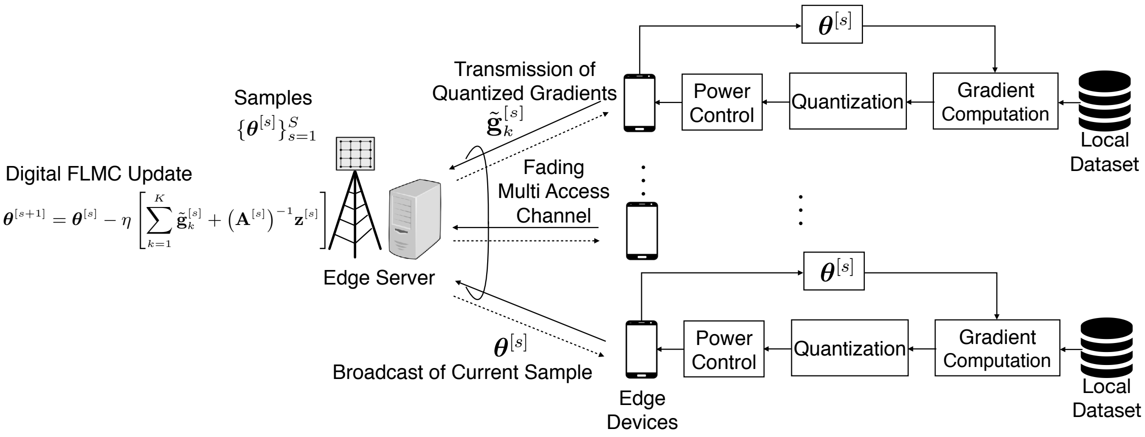
Samples
$\{\boldsymbol{\theta}^{[s]}\}_{s=1}^{S}$
Digital FLMC Update
$\boldsymbol{\theta}^{[s+1]} = \boldsymbol{\theta}^{[s]} - \eta \left[\sum_{k=1}^{K} \tilde{\mathbf{g}}_k^{[s]} + \left(\mathbf{A}^{[s]}\right)^{-1} \mathbf{z}^{[s]} \right]$
Edge Server
Transmission of Quantized Gradients
$\tilde{\mathbf{g}}_k^{[s]}$
Fading Multi Access Channel
$\boldsymbol{\theta}^{[s]}$
Broadcast of Current Sample
Edge Devices
$\boldsymbol{\theta}^{[s]}$
Power Control
Quantization
Gradient Computation
Local Dataset
$\boldsymbol{\theta}^{[s]}$
Power Control
Quantization
Gradient Computation
Local Dataset

Figure A.1 Differentially private quantized federated Bayesian learning system based on LMC.

Bayesian learning provides a more effective and principled framework to define the learning problem (see, e.g., [83]). Bayesian learning assigns a probability distribution to the model parameters, rather than collapsing any residual uncertainty in the model parameter space to a single point estimate. In this paper, we focus on the distributed implementation of Bayesian learning in wireless systems within a federated learning setting, with the main goal of leveraging the wireless channel as part of the “compute continuum” between devices and server [3] (see Fig A.1).

Scalable Bayesian learning solutions are either based on variational inference, whereby the distribution over the model parameters is optimized by minimizing a free energy metric [75]; or on Monte Carlo (MC) sampling, whereby the distribution over the model parameters is represented by random samples [7]. It was recently pointed out in [102] that MC solutions enable a novel interpretation of the wireless channel as part of the MC sampling process. In particular, reference [102] proposed a Bayesian federated learning protocol based on Langevin MC (LMC), a noise-perturbed gradient-based MC strategy [7], and *analog* transmission. The paper demonstrated the role of the channel noise as a contributor to the LMC update, as well as a privacy mechanism (see also [85, 101]). In this paper, we devise an alternative strategy that implements LMC in a federated setting via *digital* modulation under privacy constraints.

Federated learning has been widely studied for implementation on wireless channels (see, e.g., [56]). Techniques that leverage the wireless channel for computation include over-the-air computation (AirComp), whereby superposition in non-orthogonal multiple access (NOMA) is used as a means to aggregate information from different sources [128, 18, 105]; channel noise for privacy, which enforces differential privacy (DP) guarantees via power control [150, 101]; and channel noise for sampling, which was introduced above [102]. Also related to this work are DP mechanisms based on stochastic quantization [51].

In this chapter, inspired by [101], we study Bayesian federated learning protocols based on the digital transmission of gradients from edge devices to the edge server (see Fig. A.1). Like [102], which considered analog transmission, we aim at leveraging channel noise for both channel-driven MC sampling and DP. The main contributions of this paper are as follows.

- **Quantized federated LMC (FLMC):** We introduce a quantized federated implementation of LMC based on stochastic quantization, binary transmission, and channel-driven sampling;
- **Power allocation policy with DP guarantees:** We analyze the DP guarantees of LMC, and we design an approach to determine power control parameter to meet the requirements of both MC sampling and DP;
- **Experiments:** We demonstrate an experimental comparison of digital and analog wireless FLMC implementations under DP constraints.

The remainder of this chapter is organized as follows. Sec. A.3 formulates the system models and definitions. The privacy anaysis and power control design are presented in Sec. A.4. Sec. A.5 describes numerical results.

A.3 System Model

As shown in Fig. A.1, we consider a wireless federated edge learning system comprising an edge server and K edge devices. The devices are connected to the server via a shared wireless channel. Each device k has its own local dataset $\mathcal{D}_k$, which includes N_k data samples $\mathcal{D}_k = \{\mathrm{d}_{k,n}\}_{n=1}^{N_k}$. The global data set is denoted as $\mathcal{D} = \{\mathcal{D}_k\}_{k=1}^{K}$. The devices communicate to the server via a NOMA digital channel with BPSK modulation as in [212]. Unlike [212], which focuses on conventional frequentist learning, here the goal is to carry out Bayesian learning by approximating the global posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ at the server. Furthermore, as in [101], which considers analog transmission, we impose privacy constraints via DP.

A.3.1 Federated Langevin Monte Carlo

The machine learning model adopted by the system is defined by a likelihood function $p(\mathrm{d}|\boldsymbol{\theta})$, as well as by a prior distribution $p(\boldsymbol{\theta})$. Accordingly, the likelihood of the data at device k is obtained by assuming i.i.d. observations as

$$p(\mathcal{D}_k|\boldsymbol{\theta}) = \prod_{n=1}^{N_k} p(\mathrm{d}_{n,k}|\boldsymbol{\theta}). \tag{A.1}$$

The target global posterior is

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta}) \prod_{k=1}^{K} p(\mathcal{D}_k|\boldsymbol{\theta}), \tag{A.2}$$

which can be expressed in terms of the product $p(\boldsymbol{\theta}|\mathcal{D}) \propto \prod_{k=1}^{K} \tilde{p}(\boldsymbol{\theta}|\mathcal{D}_k)$ of the local sub-posteriors at each device k

$$\tilde{p}(\boldsymbol{\theta}|\mathcal{D}_k) \propto p(\boldsymbol{\theta})^{1/K} p(\mathcal{D}_k|\boldsymbol{\theta}). \tag{A.3}$$

We introduce the local cost function

$$f_k(\boldsymbol{\theta}) = -\log p(\mathcal{D}_k|\boldsymbol{\theta}) - \frac{1}{K}\log p(\boldsymbol{\theta}), \tag{A.4}$$

which accounts for prior and likelihood at device k, as well as the global cost function

$$f(\boldsymbol{\theta}) = \sum_{k=1}^{K} f_k(\boldsymbol{\theta}). \tag{A.5}$$

LMC is a gradient-based MCMC sampling scheme. As such, it aims at producing samples from the global posterior $p(\boldsymbol{\theta}|\mathcal{D})$ in (A.2) by leveraging information about the gradient of the local cost functions (A.4). At each s-th iteration, LMC produces the next sample $\boldsymbol{\theta}^{[s+1]}$ as

$$\text{(LMC)} \quad \boldsymbol{\theta}^{[s+1]} = \boldsymbol{\theta}^{[s]} - \eta \sum_{k=1}^{K} \nabla f_k(\boldsymbol{\theta}^{[s]}) + \sqrt{2\eta}\boldsymbol{\xi}^{[s+1]}, \tag{A.6}$$

where η is the step size, and $\{\boldsymbol{\xi}^{[s]}\}$ is a sequence of i.i.d. random vectors following the Gaussian distribution $\mathcal{N}(0, \mathbf{I}_m)$, which are independent of the initialization $\boldsymbol{\theta}^{[0]} \in \mathbb{R}^m$.

To implement LMC in the described federated setting, at each s-th communication round, the edge server broadcasts the current sample $\boldsymbol{\theta}^{[s]}$ to all edge devices via the downlink channel. We assume ideal downlink communication. By using the received vector $\boldsymbol{\theta}^{[s]}$ and the local dataset $\mathcal{D}_k$, each device computes the gradient of the local cost function (A.4) as

$$\mathbf{g}_k^{[s]} = -\sum_{n=1}^{N_k} \nabla \log p(\mathbf{d}_n|\boldsymbol{\theta}^{[s]}) - \frac{1}{K}\nabla \log p(\boldsymbol{\theta}^{[s]}). \tag{A.7}$$

While [101] explored the use of analog communication to transmit the local gradients in (A.7), in this work we assume that the devices apply entry-wise binary quantization in order to enable BPSK-based transmission. The edge server aggregates the received signals to obtain an approximation of the update term $-\eta \nabla f(\boldsymbol{\theta}^{[s]}) + \sqrt{2\eta}\boldsymbol{\xi}^{[s+1]}$ in (A.6). As we will see, and as first proposed in [101], channel noise can be leveraged to contribute to the additive random term $\boldsymbol{\xi}^{[s+1]}$ in the LMC update (A.6), as well as a DP mechanism. After S communication rounds, the server obtains a sequence of samples of model parameter vectors $\{\boldsymbol{\theta}^{[s]}\}_{s=1}^{S}$.

A.3.2 Communication Model

The devices communicate via NOMA on the uplink to the edge server. At any s-th communication round, each entry $\mathrm{g}_{k,i}^{[s]}$ of the gradient vector $\mathbf{g}_k^{[s]} = [\mathrm{g}_{k,1}^{[s]}, \mathcal{D}ots, \mathrm{g}_{k,m}^{[s]}]^{\mathsf{T}}$ is quantized via one-bit stochastic quantization [72]

$$\tilde{\mathrm{g}}_{k,i}^{[s]} = \begin{cases} 1 & \text{with probability } \Phi(\mathrm{g}_{k,i}^{[s]}), \\ -1 & \text{with probability } 1 - \Phi(\mathrm{g}_{k,i}^{[s]}), \end{cases} \tag{A.8}$$

where function $\Phi(\cdot)$ returns a probability that increases with the input argument. An example is given by the sigmoid function $\Phi(x) = \sigma(x) = \big(1 + \exp(-ax)\big)^{-1}$ for some fixed parameter $a > 0$. Each of the quantized gradient parameters $\tilde{\mathrm{g}}_{k,i}^{[s]}$ is modulated into one BPSK symbol. As a result, a block of m BPSK symbols is produced to communicate the quantized local gradient vector $\tilde{\mathbf{g}}_k^{[s]}$ in a communication round.

Accordingly, at the s-th communication round, the received signal at the server is given by the superposition

$$\mathbf{y}^{[s]} = \sum_{k=1}^{K} \mathbf{H}_k^{[s]} \mathbf{P}_k^{[s]} \tilde{\mathbf{g}}_k^{[s]} + \mathbf{z}^{[s]}, \tag{A.9}$$

where $\mathbf{H}_k^{[s]} = \text{diag}[h_{k,1}^{[s]}, \cdots, h_{k,m}^{[s]}]$ and $\mathbf{P}_k^{[s]} = \text{diag}[P_{k,1}^{[s]}, \cdots, P_{k,m}^{[s]}]$ are diagonal matrices collecting respectively the channel gains and power control parameters for m consecutive symbols in a block; while $\mathbf{z}^{[s]}$ is the channel noise, which is i.i.d. according to distribution $\mathcal{N}(0, N_0\mathbf{I})$. We assume perfect channel state information (CSI) at all nodes, so that, as we will see, each device can compensate for the phase and amplitude of its own channel.

In the following sections, we will design the power allocation parameters $\{\{P_{k,i}^{[s]}\}_{i=1}^{m}\}_{k=1}^{K}$ for each communication round. The transmission of each device is subject to the average per block transmission power constraint:

$$\text{(Power constraint)} \quad \frac{1}{m}\sum_{i=1}^{m} \left|P_{k,i}^{[s]}\tilde{\mathrm{g}}_{k,i}^{[s]}\right|^2 \leq P_0, \forall k, s. \tag{A.10}$$

We define the maximum signal to noise ratio (SNR) as $\text{SNR}_{\mathsf{max}} = P_0/N_0$, which is obtained when a device transmits at full power.

A.3.3 Differential Privacy

We assume an “honest-but-curious" edge server that may attempt to infer information about local data sets from the received signals $\mathbf{y}^{[s]}$. The privacy constraint is described by the standard (ϵ,δ)-DP constraint, with some $\epsilon>0$ and $\delta\in[0,1)$. DP hinges on the divergence between the two distributions $P(\mathbf{y}^{[s]}|\mathcal{D}')$ and $P(\mathbf{y}^{[s]}|\mathcal{D}'')$ of the signal received when the data sets $\mathcal{D}'$ and $\mathcal{D}''$ differ a single data point, i.e., $\|\mathcal{D}'-\mathcal{D}''\|_1=1$. Formally, we have (ϵ,δ)-DP if the inequality

$$\max_{\mathcal{D}',\mathcal{D}'':\|\mathcal{D}'-\mathcal{D}''\|_1=1}\left\{\Pr(|\mathcal{L}_{\mathcal{D}',\mathcal{D}''}(\mathbf{y}^{[s]})|\le\epsilon)\right\}\ge 1-\delta \tag{A.11}$$

is satisfied, where the DP loss $\mathcal{L}_{\mathcal{D}',\mathcal{D}''}(\mathbf{y}^{[s]})$ is

$$\mathcal{L}_{\mathcal{D}',\mathcal{D}''}(\mathbf{y}^{[s]})=\log\frac{P(\mathbf{y}^{[s]}|\mathcal{D}')}{P(\mathbf{y}^{[s]}|\mathcal{D}'')}. \tag{A.12}$$

The probability in (A.11) is taken with respect to the distribution $P(\mathbf{y}^{[s]}|\mathcal{D}')$. We note that the DP constraint (A.11) is applied at each communication round, and that the overall privacy guarantees across iterations can be obtained by using standard composition theorems [36, Sec. 3.5]. To ensure DP requirement as [24, 187], we make the following assumption on the gradients.

Assumption 2 (Bounded Gradients)**.** Each element of the local gradients is bounded by some constant $\ell>0$ as

$$\left|\mathrm{g}_{k,i}^{[s]}\right|\le\ell,\quad \text{for all } k,s,i. \tag{A.13}$$

In practice, the condition (A.13) can be met by clipping each entry of the gradient as $\min\{1,\ell/|\mathrm{g}_{k,i}^{[s]}|\}\mathrm{g}_{k,i}^{[s]}$ before quantization [24].

A.4 Power Control for Quantized Federated Langevin Monte Carlo

In this section, we first present the transmitter and receiver designs for the proposed quantized federated Langevin Monte Carlo (FLMC), and then analyze its DP properties. Finally, we address the design of power control parameters in (A.9).

A.4.1 Signal Design

As described in Sec. A.3.2, each device applies stochastic quantization as in (A.8). Followed by BPSK transmission under the assumption of perfect CSI, we consider channel inversion, whereby the power control matrix in (A.9) is selected as $\mathbf{P}_k^{[s]} = \mathbf{A}^{[s]}(\mathbf{H}_k^{[s]})^{-1}$. The diagonal matrix $\mathbf{A}^{[s]} = \mathrm{diag}[A_1^{[s]}, \cdots, A_m^{[s]}]$ is to be designed with the goal of ensuring that the server can approximate the LMC update (A.6), while also guaranteeing the power constraint (A.10) and the DP constraint (A.11).

The server normalizes the received signal as $(\mathbf{A}^{[s]})^{-1}\mathbf{y}^{[s]}$ to obtain an estimate of the global gradient. Accordingly, the server approximates the LMC update (A.6) as

$$\boldsymbol{\theta}^{[s+1]} = \boldsymbol{\theta}^{[s]} - \eta \left[\sum_{k=1}^{K} \tilde{\mathbf{g}}_k^{[s]} + \left(\mathbf{A}^{[s]}\right)^{-1} \mathbf{z}^{[s]} \right]. \tag{A.14}$$

A.4.2 Privacy Analysis

We now consider the DP constraint (A.11) for any device k. To this end, we fix the quantized gradients $\{\tilde{\mathbf{g}}_j\}_{j \neq k}$ of the other devices, and consider neighboring data sets $\mathcal{D}_k'$ and $\mathcal{D}_k''$ for device k that differ only by one sample, i.e., $\|\mathcal{D}_k' - \mathcal{D}_k''\|_1 = 1$. As the DP constraint (A.11) is applied to every iteration, we omit the index of the communication round s for ease of notation. Then, the privacy loss (A.12) for device k can be written as

$$\begin{aligned} \mathcal{L}_{\mathcal{D}',\mathcal{D}''}(\mathbf{y}) &= \log \frac{\prod_{i=1}^{m} P(A_i \tilde{\mathrm{g}}_{k,i}' + A_i \sum_{q \neq k} \tilde{\mathrm{g}}_{q,i} + z_i \big| \{\tilde{\mathrm{g}}_{q,i}\}_{q \neq k}, \mathcal{D}_k')}{\prod_{i=1}^{m} P(A_i \tilde{\mathrm{g}}_{k,i}'' + A_i \sum_{q \neq k} \tilde{\mathrm{g}}_{q,i} + z_i \big| \{\tilde{\mathrm{g}}_{q,i}\}_{q \neq k}, \mathcal{D}_k'')} \\ &= \sum_{i=1}^{m} \log \frac{\left[\Phi(\mathrm{g}_{k,i}') \exp\left(\frac{2(z_i - A_i \sum_{q \neq k} \tilde{\mathrm{g}}_{q,i})}{N_0/A_i} \right) + \left(1 - \Phi(\mathrm{g}_{k,i}') \right) \right]}{\left[\Phi(\mathrm{g}_{k,i}'') \exp\left(\frac{2(z_i - A_i \sum_{q \neq k} \tilde{\mathrm{g}}_{q,i})}{N_0/A_i} \right) + \left(1 - \Phi(\mathrm{g}_{k,i}'') \right) \right]}, \end{aligned} \tag{A.15}$$

where, with some abuse of notation, $P(X|Y)$ represents the distribution of random variable X evaluated at X when conditioned on the value Y of random variable Y; the last step uses the fact that the distributions in (A.15) are mixture of Gaussians; and we have $z_i \sim \mathcal{N}(0, N_0)$. To attain the maximum DP loss in (A.15), we consider the worst-case choice of data sets $\mathcal{D}'$ and $\mathcal{D}''$. To this end, without loss of generality, we set $\Phi(\mathrm{g}_{k,i}') = \Phi(\ell)$ and $\Phi(\mathrm{g}_{k,i}'') = \Phi(-\ell)$ by Assumption 2. Furthermore, the value of the sum $\sum_{j \neq k} \tilde{\mathrm{g}}_{q,i}$ is within the range of $[-(K-1),(K-1)]$, and hence have the following inequality

$$\begin{aligned}
|\mathcal{L}_{\mathcal{D}',\mathcal{D}''}(\mathbf{y})| \leq \max\Bigg\{ & \left| \sum_{i=1}^{m} \log \frac{\left[\Phi(\ell)\exp\left(\frac{2(z_i+A_i(K-1))}{N_0/A_i}\right)+\left(1-\Phi(\ell)\right)\right]}{\left[\Phi(-\ell)\exp\left(\frac{2(z_i+A_i(K-1))}{N_0/A_i}\right)+\left(1-\Phi(-\ell)\right)\right]}\right|, \\
& \left| \sum_{i=1}^{m} \log \frac{\left[\Phi(\ell)\exp\left(\frac{2(z_i-A_i(K-1))}{N_0/A_i}\right)+\left(1-\Phi(\ell)\right)\right]}{\left[\Phi(-\ell)\exp\left(\frac{2(z_i-A_i(K-1))}{N_0/A_i}\right)+\left(1-\Phi(-\ell)\right)\right]}\right| \Bigg\} \\
& \triangleq \mathcal{L}^*(\mathbf{z}),
\end{aligned} \tag{A.16}$$

where $\mathbf{z} \sim \mathcal{N}(0,\mathbf{I}_m)$. We can now use (A.16) to evaluate numerically a bound on left-hand side of (A.11) as $\Pr(|\mathcal{L}^*(\mathbf{z})| \leq \epsilon) \geq 1-\delta$ with $\mathbf{z} \sim \mathcal{N}(0,\mathbf{I}_m)$.

To compare with analog FLMC in [101], we reproduce the privacy loss in [101] as

$$\mathcal{L}_{\mathcal{D}',\mathcal{D}''}(\mathbf{y}) = \sum_{i=1}^{m} \frac{2z_i A_i \Delta_{k,i} + (A_i \Delta_{k,i})^2}{2N_0}, \tag{A.17}$$

where $z_i \sim \mathcal{N}(0,N_0)$, and $\Delta_{k,i} = |\mathrm{g}'_{k,i} - \mathrm{g}'_{k,i}|$, and we have $\Delta_{k,i} \leq 2\ell$. To gain some insight about the comparison between (A.16) and (A.17), consider the high-SNR regime in which the power of channel noise N_0 approaches 0. In this case, the privacy loss (A.17) in the analog scheme goes to infinity, and hence no (ϵ,δ)-DP level with $\delta < 1$ is possible. This is in sharp contrast with the digital scheme, for which the privacy loss (A.16) is upper bounded by $m\log\Phi(\ell) - m\log\Phi(-\ell)$. This discussion illustrates the potential advantages of the digital scheme in the presence of privacy constraints in the high-SNR regime.

A.4.3 Power Control

The design of power control parameters in the power gain matrix $\mathbf{A}^{[s]}$ must comply with the power constraints, the LMC noise requirements, and the DP constraints.

For the power constraint (A.10), plugging in the choice $\mathbf{P}_k^{[s]} = \mathbf{A}^{[s]}(\mathbf{H}_k^{[s]})^{-1}$ yields the inequalities

$$\frac{1}{m}\sum_{i=1}^{m}\left(\frac{A_i^{[s]}}{h_{k,i}^{[s]}}\right)^2 \leq P_0,\ \forall k,s. \tag{A.18}$$

Furthermore, in order to guarantee that the noise powers $N_0\eta^2(A_i^{[s]})^{-2}$ in the update (A.14) are no smaller than the power 2η required by the LMC update (A.6) we impose the

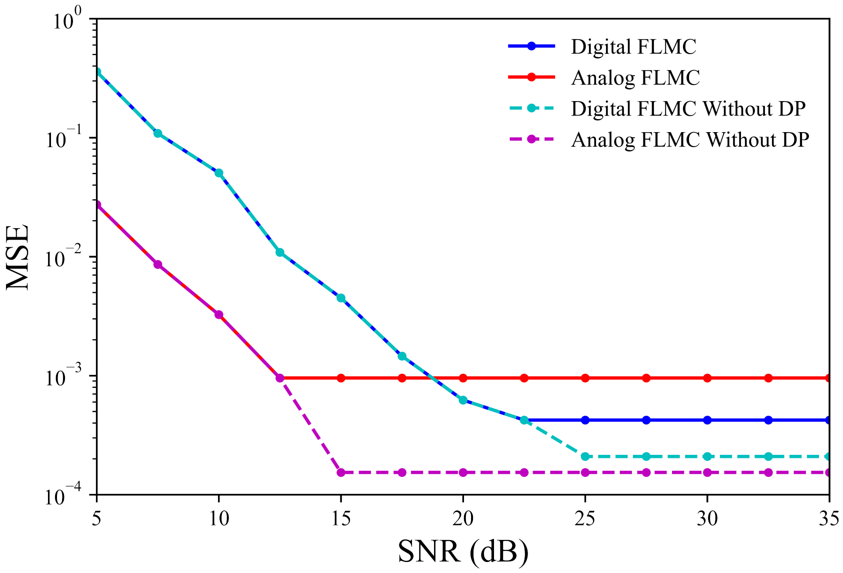

Figure A.2 MSE as a function of SNR ($\epsilon = 5, \delta = 0.01$).

LMC noise requirement (see also [102])

$$A_i^{[s]} \leq \sqrt{\frac{\eta N_0}{2}}, \ \forall i, s. \tag{A.19}$$

Finally, to impose the DP constraint, given the desired level of privacy loss ϵ, we numerically estimate the probability δ in (A.11) as a function of power gain parameters $A_i^{[s]}$ by drawing samples from the noise $\mathbf{z}^{[s]} \sim \mathcal{N}(0, N_0\mathbf{I})$.

A.5 Numerical Results

In this section, we evaluate the performance of the proposed quantized FLMC, and compare it with the analog transmission scheme introduced in [102]. Throughout this section, we assume the channel coefficients to be constant within a communication block, and homogeneous across the devices, i.e., $h_{k,i}^{[s]} = h^{[s]}$ for all devices $k = 1, \dots, K$ and all elements $i = 1, \dots, m$. Under this assumption, the power gains for quantized FLMC are obtained via a numerical search to maximize the value of $A_i^{[s]}$ under the three constraints reviewed in the previous sections. In a similar manner, for **analog FLMC**, we have [101]

$$A_i^{[s]} = \min\left\{\frac{|h^{[s]}|\sqrt{P_0}}{\ell}, \sqrt{\frac{\eta N_0}{2}}, \sqrt{\frac{N_0 \mathcal{T}^{-1}(1-\delta)}{2m\ell^2}}\right\}, \ \forall k, \ s, \tag{A.20}$$

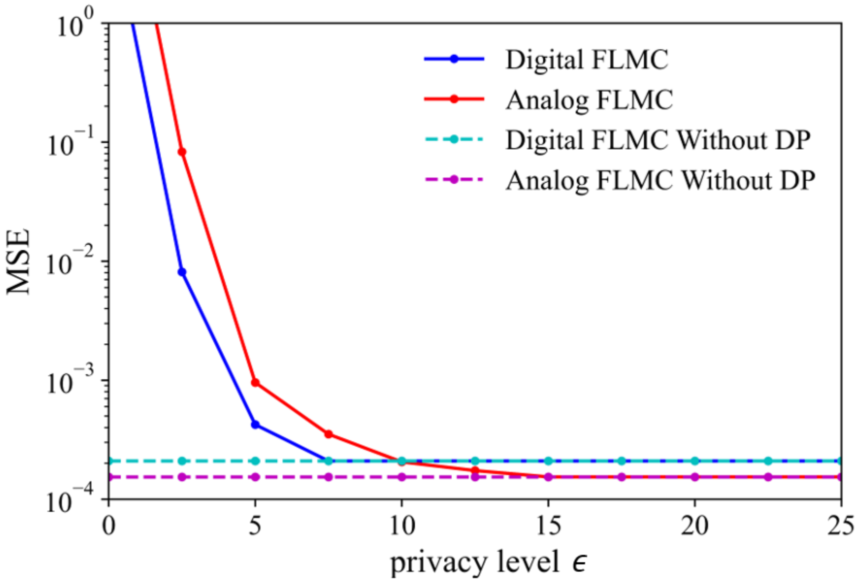

Figure A.3 MSE as a function of privacy level ϵ ($\text{SNR}_{\text{max}} = 25$ dB, $\delta = 0.01$).

where the last square root term is the inverse function of $\mathcal{T}(x)$ defined by the error function $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$, denoted as

$$\mathcal{T}(x) = \text{erf}\left(\frac{\epsilon - x}{2\sqrt{x}}\right) - \text{erf}\left(\frac{-\epsilon - x}{2\sqrt{x}}\right), \tag{A.21}$$

which is obtained by plugging (A.17) into (A.11), and leveraging the tail probability of Gaussian distribution. We also consider benchmark schemes without DP constraint.

As for the learning model, as in [102], we consider a Gaussian linear regression with likelihood

$$p(v_n|\boldsymbol{\theta}, \mathbf{u}_n) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(v_n - \boldsymbol{\theta}^\mathsf{T}\mathbf{u}_n)^2}, \tag{A.22}$$

and the prior $p(\boldsymbol{\theta})$ is assumed to follow Gaussian distribution $\mathcal{N}(0, \mathbf{I}_m)$. Therefore, the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ is the Gaussian $\mathcal{N}\big((\mathbf{U}\mathbf{U}^\mathsf{T} + \mathbf{I})^{-1}\mathbf{U}\mathbf{v}, (\mathbf{U}\mathbf{U}^\mathsf{T} + \mathbf{I})^{-1}\big)$, where $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_N]$ is the data matrix and $\mathbf{v} = [v_1, \cdots, v_N]^\mathsf{T}$ is the label vector. We use synthetic dataset $\{\mathbf{d}_n = (\mathbf{u}_n, v_n)\}_{n=1}^N$ with $N = 1200$ following the learning model in (A.22), with input $\mathbf{u}_n$ drawn i.i.d from $\mathcal{N}(0, \mathbf{I}_m)$ where $m = 5$. The ground-truth model parameter is $\boldsymbol{\theta}^* = [0.418, -0.289, 0.3982, 0.8231, 0.5251]^\mathsf{T}$. Unless stated otherwise, the data set is evenly distributed to $K = 20$ devices; the constant channel $h^{[s]}$ is set to 0.04 for all communication rounds; the power of channel noise is set to $N_0 = 1$; the bound of gradient element is set to $\ell = 30$; learning rate is set to $\eta = 1.28 \times 10^{-4}$ for analog FLMC and $\eta = 8.28 \times 10^{-3}$ for digital FLMC, which are tuned by using the smoothness and

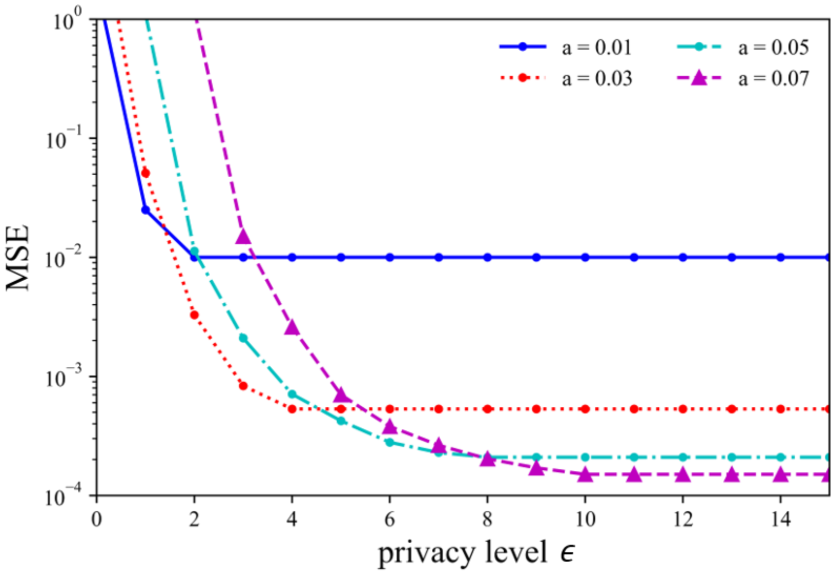

Figure A.4 MSE as a function of privacy level ϵ for different parameter of the stochastic binary quantization a (SNR$_{\mathsf{max}} = 25$ dB, $\delta = 0.01$).

strongly convexity parameters (see [102]). We consider a sigmoid function for quantization probability in (A.8) as $\Phi(x) = [1 + \exp(-ax)]^{-1}$, and set $a = 0.05$ by default.

The total number of communication rounds is chosen as $S = 300$, which are comprised of $S_b = 200$ samples for the burn-in period, and the following $S_u = S - S_b = 100$ samples for evaluation. The quality of the samples is measured by mean squared error (MSE)

$$\text{MSE} = \frac{1}{S_u} \sum_{s=S_b+1}^{S_b+S_u} \|\boldsymbol{\theta}^{[s]} - \boldsymbol{\mu}\|^2, \tag{A.23}$$

where $\boldsymbol{\mu}$ is the mean of the ground-truth posterior distribution. All the results are averaged over 1000 experiments.

We first investigate the impact of SNR in Fig. A.2 on the performance of digital and analog FLMC schemes. In this experiment, we set the DP level as $\epsilon = 5$ and $\delta = 0.01$. Confirming the discussion in the previous section, in the high-SNR regime, digital FLMC is seen to outperform analog FLMC, since the latter one must back off the transmitted power in order to meet the DP constraint. In contrast, SNR lower than 17.5 dB, analog FLMC is preferable.

We now further investigate the impact of the privacy level on the digital and analog FLMC schemes in Fig. A.3. In this experiment, we set SNR$_{\mathsf{max}} = 25$ dB. The error of all schemes is seen to decrease by relaxing the DP constraint, until $\epsilon = 7.5$ for the digital scheme and $\epsilon = 15$ for the analog scheme. Relaxing the DP constraint cannot reduce the error, as the performance becomes limited by the transmitted power constraint or by LMC noise requirement. The digital FLMC scheme outperforms analog FLMC under a stricter

DP requirement, i.e., when $\epsilon \leq 7.5$. This provides further validation of the advantage of the digital scheme when the SNR is large enough.

Finally, in Fig. A.4, we study the impact of varying the parameter a of the quantization probability function $\Phi(x) = [1 + \exp(-ax)]^{-1}$. Note that a small a implies a more noisy quantizer. In this experiment, we also set $\mathsf{SNR}_{\mathsf{max}} = 25$ dB. Under strict DP requirement $\epsilon < 2$, the quantizer with the small value $a = 0.01$ outperforms other choices, since the higher level of randomness is applied to meet the DP constraint. Conversely, by relaxing the DP requirement, quantizer with larger value of a become advantageous.

Appendix B

Supplementary Materials for Chapter 2

B.1 Approximation Schemes for Gibbs Hyper-posterior

In this section, we describe two tractable schemes for approximating the Gibbs hyperposterior (3.16).

Maximum A Posteriori (MAP) Estimate: The MAP estimate approximates the Gibbs hyperposterior $q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N})$ by a Dirac measure centered at its mode

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N}). \tag{B.1}$$

The mode $\boldsymbol{\theta}^*$ can be evaluated equivalently as the solution to the following optimization problem,

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left\{ -\log p(\boldsymbol{\theta}) + \gamma^{-1}\bar{\mathcal{L}}(\boldsymbol{\theta},\mathcal{D}_{1:N}) \right\}. \tag{B.2}$$

With the choice of a zero mean isotropic Gaussian distribution as the hyper-prior, i.e., $p(\boldsymbol{\theta}) \sim \mathcal{N}(0,\sigma_{\mathcal{P}}^2 I)$, (B.2) results in

$$\arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} J^{\text{MAP}}(\boldsymbol{\theta}) = \gamma^{-1}\bar{\mathcal{L}}(\boldsymbol{\theta},\mathcal{D}_{1:N}) + \frac{1}{2\sigma_{\mathcal{P}}^2}||\boldsymbol{\theta}||_2^2, \tag{B.3}$$

where the objective $J^{\text{MAP}}(\boldsymbol{\theta})$ consists of a sum of meta-training loss (3.12) and an $L2$-regularization term.

To optimize $J^{\text{MAP}}(\boldsymbol{\theta})$, we use mini-batch gradient descent in training, where the mini-batches are sampled at meta-level (i.e., we sample mini-batches of tasks and use all data points of the corresponding tasks to compute the gradients of $J^{\text{MAP}}(\boldsymbol{\theta})$).

Stein Variational Gradient Descent (SVGD): SVGD is a general purpose variational inference algorithm that aims to minimize the KL divergence, $D(\hat{q}(\boldsymbol{\theta})||q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N}))$, to the target distribution $q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N})$, over non-parametric distributions $\hat{q}(\boldsymbol{\theta})$. The distribution $\hat{q}(\boldsymbol{\theta})$ is represented by a collection of particles $\{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,...,\boldsymbol{\theta}_K\}$, which can in turn be used to approximate $\hat{q}(\boldsymbol{\theta})$ via a Kernel Density Estimator (KDE) [14].

For the SVGD approximation of $q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N})$, we start by sampling K particles $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_K\}$ from the hyper-prior $p(\boldsymbol{\theta})$. The particles are then iteratively transported to minimize the KL-divergence $D(\hat{q}(\boldsymbol{\theta})||q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N}))$ via a form of functional gradient descent on a reproducing kernel Hilbert space (RKHS), induced by a kernel function $\tilde{k}(\cdot,\cdot)$. Specifically, we choose a squared exponential kernel,

$$\tilde{k}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \exp\left(-\frac{||\boldsymbol{\theta} - \boldsymbol{\theta}'||_2^2}{2l}\right), \tag{B.4}$$

with l denoting the fixed, length hyperparameter. Consequently, the SVGD update at iteration t is given as

$$\boldsymbol{\theta}_k^{[t]} \leftarrow \boldsymbol{\theta}_k^{[t-1]} + \epsilon\Phi(\boldsymbol{\theta}_k^{[t-1]}), \tag{B.5}$$

where

$$\begin{aligned}\Phi(\boldsymbol{\theta}_k^{[t-1]}) =& \frac{1}{K}\sum_{j=1}^{K}\Big[\tilde{k}(\boldsymbol{\theta}_j^{[t-1]}, \boldsymbol{\theta}_k^{[t-1]})\nabla_{\boldsymbol{\theta}_j^{[t-1]}} \log q^{\text{WFEM-GP}}(\boldsymbol{\theta}_j^{[t-1]}|\mathcal{D}_{1:N}) \\ &+ \nabla_{\boldsymbol{\theta}_j^{[t-1]}}\tilde{k}(\boldsymbol{\theta}_j^{[t-1]}, \boldsymbol{\theta}_k^{[t-1]})\Big]\end{aligned} \tag{B.6}$$

for each particle $k = 1, \ldots, K$. Moreover, it has been shown in [104] that in the asymptotic limit as $K \to \infty$, the empirical distribution encoded by the particles $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K\}$ converges to the true target distribution $q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N})$.

To estimate the score function $\nabla_{\boldsymbol{\theta}_j^{[t-1]}} \log q^{\text{WFEM-GP}}(\boldsymbol{\theta}_j^{[t-1]}|\mathcal{D}_{1:N})$, we use a mini-batch of n meta-training datasets with $\mathcal{D}_1, \ldots, \mathcal{D}_{\beta n}$ data from source environment data set and $\mathcal{D}_{\beta n+1}, \ldots, \mathcal{D}_n$ from target environment dataset. Using this, the score function is approximated as

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_j^{[t-1]}} \log q^{\text{WFEM-GP}}(\boldsymbol{\theta}_j^{[t-1]}|\mathcal{D}_{1:N}) \approx& \frac{N}{n}\gamma\Big[\alpha\sum_{i=1}^{\beta n}\frac{1}{m_i}\nabla_{\boldsymbol{\theta}_j^{[l-1]}} \log p_{\boldsymbol{\theta}_j^{[l-1]}}(\mathbf{y}_i|\mathbf{X}_i) \\ &+ (1-\alpha)\sum_{i=\beta n+1}^{n}\frac{1}{m_i}\nabla_{\boldsymbol{\theta}_j^{[t-1]}} \log p_{\boldsymbol{\theta}_j^{[t-1]}}(\mathbf{y}_i|\mathbf{X}_i)\Big] \\ &+ \nabla_{\boldsymbol{\theta}_j^{[t-1]}} \log p(\boldsymbol{\theta}_j^{[t-1]}).\end{aligned} \tag{B.7}$$

B.2 Meta-Testing of WFEM-GP under MAP and SVGD

During meta-testing, the hyperparameter meta-learned using WFEM-GP is used to initialize the GP prior of a meta-test task. The GP is subsequently fitted to the meta-test training data set $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ of M samples to yield a GP posterior. We assume that a test data set, $\mathcal{D}^* = \{(\mathbf{x}_m^*, y_m^*)_{m=1}^M\}$, is also available, independent of the training data set $\mathcal{D}$. The performance of the GP posterior for the meta-test task is then evaluated on $\mathcal{D}^*$.

To this end, corresponding to each test input $\mathbf{x}_m^*$, for $m = 1,\ldots,M$, we evaluate the average predictive posterior distribution $\mathbb{E}_{q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N})}[p_{\boldsymbol{\theta}}(f(\mathbf{x}_m^*)|\mathcal{D})]$ defined in (3.11). Under the MAP scheme, which yields a point estimate $\boldsymbol{\theta}^*$ of the Gibbs hyper-posterior $q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N})$, the average predictive posterior distribution corresponds to $p_{\boldsymbol{\theta}^*}(f(\mathbf{x}_m^*)|\mathcal{D})$. In contrast, SVGD scheme outputs K particles, $\boldsymbol{\theta}_k \sim q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N})$, for $k = 1,\ldots,K$. These can be used to approximate the average predictive posterior distribution as

$$\mathbb{E}_{\boldsymbol{\theta}\sim q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N})}[p_{\boldsymbol{\theta}}(f(\mathbf{x}_m^*)|\mathcal{D})] \approx \frac{1}{K}\sum_{k=0}^{K} p_{\boldsymbol{\theta}_k}(f(\mathbf{x}_m^*)|\mathcal{D}). \tag{B.8}$$

It is easy to see that for $K = 1$ and $\boldsymbol{\theta}_k = \boldsymbol{\theta}^*$, the SVGD scheme coincides with MAP.

To evaluate the predictive performance for regression experiment, we use average root mean square error (RMSE) as the metric, which can be computed as follows. For each $\boldsymbol{\theta}_k$ in the SVGD scheme (or $\boldsymbol{\theta}^*$ for MAP scheme), we consider the mean prediction of $p_{\boldsymbol{\theta}_k}(f(\mathbf{x}_m^*)|\mathcal{D})$ as $\hat{y}(\boldsymbol{\theta}_k, \mathbf{x}_m^*) = \mathbb{E}_{p_{\boldsymbol{\theta}_k}(f(\mathbf{x}_m^*)|\mathcal{D})}[f(\mathbf{x}^*)]$. Subsequently, the mean prediction corresponding to (B.8) evaluates as

$$\hat{y}(\mathbf{x}_m^*) = \frac{1}{K}\sum_{k=0}^{K} \hat{y}(\boldsymbol{\theta}_k, \mathbf{x}_m^*). \tag{B.9}$$

The average root mean squared error (RMSE) is then computed as

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{m=1}^{M} (\hat{y}(\mathbf{x}_m^*) - y_m^*)^2}. \tag{B.10}$$

In our experiments, we set the number of SVGD particles to be $K = 10$. To evaluate the predictive performance for classification, we adopt mean accuray as the metric, which can be computed as follows. In each dataset, we sum the absolute difference between prediction $\hat{y}(\mathbf{x}_m^*)$ and label y_m^*, then compute the mean accuracy as

$$\text{MeanAccuracy} = 1 - \frac{1}{M}\sum_{m=1}^{M} |\hat{y}(\mathbf{x}_m^*) - y_m^*|. \tag{B.11}$$

B.3 Laplace Approximation-Based GP Binary Classifier

In this section, we review the Laplace approximation based implementation of the GP binary classifier [142] which trains on an input data set $\mathcal{D} = (\mathbf{X}, \mathbf{y})$. We outline the key steps here and refer the readers to [142] for more details.

For the binary classification problem using GP, we assume a logistic regression model with $p(y = +1|f(\mathbf{x})) = \sigma(f(\mathbf{x}))$, where $\sigma(a) = (1 + \exp(-a))^{-1}$ is the sigmoid function. Note that the function $f(\mathbf{x})$ acts as a latent function in describing the likelihood $p(y = +1|f(\mathbf{x}))$: inferring $f(\mathbf{x})$ does not yield the required predictions as in the regression problem, but has to be combined with a deterministic sigmoid function.

In GP classification, the latent function $f(\cdot)$ is assumed to be random and endowed with a GP prior. As such, corresponding to an observed input data set $\mathbf{X}$, the distribution $p_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{X}))$ defines a GP prior over the output of the latent function $\mathbf{f}(\mathbf{X})$, and

$$p(\mathbf{y}|\mathbf{f}(\mathbf{X})) = \prod_{m=1}^{M} \sigma(f(\mathbf{x}_m))^{y_m}(1 - \sigma(f(\mathbf{x}_m)))^{(1-y_m)}. \tag{B.12}$$

In contrast to GP regression, the above likelihood is non-Gaussian. The GP posterior resulting from fitting the data set $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ on the GP prior is then obtained as

$$p_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{X})|\mathcal{D}) = \frac{p(\mathbf{y}|\mathbf{f}(\mathbf{X}))p_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{X}))}{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{X})}. \tag{B.13}$$

Inference is done in two steps: in the first step, we evaluate the distribution of the output of the latent function $f(\mathbf{x})$ with respect to a test input $\mathbf{x}$ as

$$p_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathcal{D}) = \int p_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathbf{f}(\mathbf{X}))p_{\boldsymbol{\theta}}(\mathbf{f}(\mathcal{X})|\mathcal{D})df, \tag{B.14}$$

where the conditional distribution $p_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathbf{f}(\mathbf{X}))$ is Gaussian and can be evaluated directly as in [142, Equation 2.19]. The distribution of the test output of the latent function is subsequently used to make prediction as

$$p(y = 1|\mathcal{D}, \mathbf{x}) = \int \sigma(f(\mathbf{x}))p_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathcal{D})df. \tag{B.15}$$

The integrals in (B.14) and (B.15) are intractable due to the non-Gaussianity of the data likelihood. To tackle this, we use a Laplace approximation based classifier obtained via the following steps.

Approximation of the posterior $p_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{X})|\mathcal{D})$: The first step is to replace the posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{X})|\mathcal{D})$ in (B.14), which is defined as in (B.13), using a Laplace approximation [14],

$$q_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{X})|\mathcal{D}) \sim \mathcal{N}(\hat{f}, \Sigma^{-1}), \tag{B.16}$$

where $\hat{f} = \arg\max_{f(\mathbf{x})} p_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{X})|\mathcal{D})$ is the mode of the posterior distribution, and

$$\Sigma = -\nabla^2 \log p_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{X})|\mathcal{D})|_{\mathbf{f}(\mathbf{X})=\hat{t}} \tag{B.17}$$

is the Hessian of the negative of the log posterior evaluated at $\hat{f}$.

Computing the mode $\hat{f}$ of the posterior distribution amounts to solving the following equation,

$$\log p(\mathbf{y}|\mathbf{f}(\mathbf{X})) - \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})^{-1}\hat{f} = 0, \tag{B.18}$$

which cannot be directly solved. As such, $\hat{f}$ is obtained using Newton method which iteratively updates an estimate $\tilde{f}$ of the mode as

$$\tilde{f} \leftarrow (\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})^{-1} + \mathbf{W})^{-1}(\mathbf{W}\tilde{f} + \nabla \log p(\mathbf{y}|\mathbf{f}(\mathbf{X}))|_{\tilde{f}}). \tag{B.19}$$

Here, $\mathbf{W} = -\nabla^2 \log p(\mathbf{y}|\mathbf{f}(\mathbf{X}))|_{\tilde{f}}$ is the Hessian of the negative log-likelihood evaluated at $\tilde{f}$. The covariance matrix of Laplace approximation $q_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{X})|\mathcal{D})$ in (B.16) corresponds to $\Sigma = (\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})^{-1} + \mathbf{W})$.

Approximation of Distribution $p_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathcal{D})$: In the next step, we approximate the test output distribution $p_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathcal{D})$ by $q_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathcal{D})$, which is obtained by replacing the posterior $p_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathcal{D})$ in (B.14) with its Laplace approximation $q_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathcal{D})$ obtained in (B.16). This yields that

$$q_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathcal{D}) = \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x})), \tag{B.20}$$

where its mean and covariance functions are denoted as

$$\mu(\mathbf{x}) = \mu_{\boldsymbol{\theta}}(\mathbf{x}) + \mathbf{k}_{\mathcal{D}}(\mathbf{x})^{\mathsf{T}} \nabla \log p(\mathbf{y}|\mathbf{f}(\mathbf{X})), \tag{B.21}$$

$$\sigma^2(\mathbf{x}) = k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{\mathcal{D}}(\mathbf{x})^{\mathsf{T}}(\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}) + \mathbf{W}^{-1})^{-1}\mathbf{k}_{\mathcal{D}}(\mathbf{x}). \tag{B.22}$$

Computation of Predictive Distribution $p(y = 1|\mathcal{D}, \mathbf{x})$: Using all the approximations explained above, the predictive distribution $p(y = 1|\mathcal{D}, \mathbf{x})$ is finally computed as

$$p(y = 1|\mathcal{D}, \mathbf{x}) \approx \int \sigma(f(\mathbf{x})) q_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathcal{D}) df. \tag{B.23}$$

The above integral is evaluated by sampling multiple latent function outputs $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_R) \sim q_{\boldsymbol{\theta}}(f(\mathbf{x})|\mathcal{D})$ and computing an equally weighted average. The predictive distribution computed as above is used to evaluate the predictive performance via RMSE.

Log-marginal Likelihood $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{X})$: Lastly, we consider a Laplace approximation to the log of the marginal likelihood $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{X})$ as

$$\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\hat{f}\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})^{-1}\hat{f} + \log p(\mathbf{y}|\mathbf{f}(\mathbf{X})) - \frac{1}{2}\log|\mathbf{I}_M + \mathbf{W}^{\frac{1}{2}}\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})\mathbf{W}^{\frac{1}{2}}|, \tag{B.24}$$

where $\mathbf{I}_M$ is the identity matrix of size M. This is required to evaluate the weighted meta-training loss (3.12) which in turn determines the Gibbs hyperposterior $q^{\text{WFEM-GP}}(\boldsymbol{\theta}|\mathcal{D}_{1:N})$.

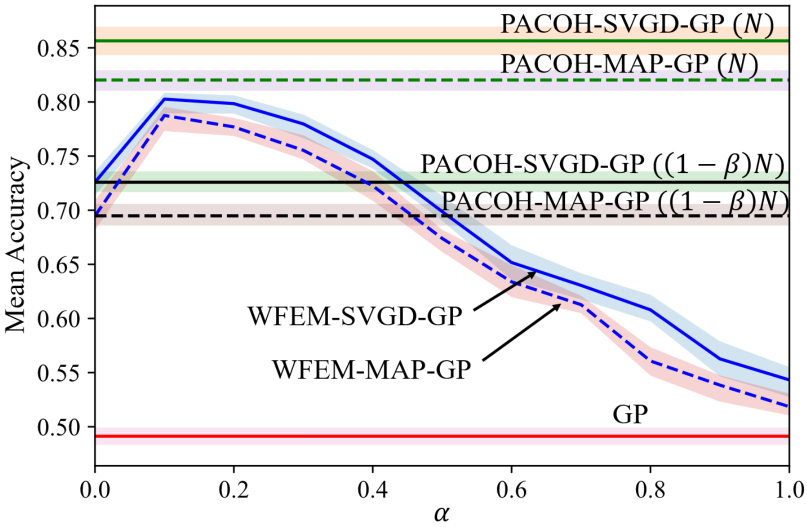

Figure B.1 Comparison of mean accuracy between MAP and SVGD of GP, PACOH-GP with N tasks and with $(1-\beta)N$ tasks from the target environment and WFME-GP against varying weight parameter α of the meta-training loss.

We evaluate the performance of the proposed transfer meta-learner in (3.16) using standard few-shot classification datasets, namely mini-ImageNet serving as source task environment and CUB for the target task environment. The mini-ImageNet is composed of 100 classes selected from ImageNet randomly, and each class has 600 images, which are resized to 84×84 pixels for fast training and inference [181]. CUB is composed of 11788 images over 200 birds classes, which are also resized to 84×84 pixels [183].

We conduct 2-way 5-shot binary classification experiments based on above datasets. Precisely, the data set for each task from the source task environment (mini-ImageNet) is obtained by first selecting 2 classes at random, and then randomly sampling 5 images for each class from mini-ImageNet dataset. The training data set from target task environment is similarly chosen from the CUB data set. For testing tasks, we sample randomly 15 images from each class.

In Fig. B.1, we compare the performance of WFEM-GP with the three benchmark schemes - GP, PACOH-GP with $(1-\beta)N$ tasks and with N tasks from target task environment – as a function of α. We plot the performances under both MAP and SVGD schemes. Other parameters are set as $N = 20$, $M_i = 5$ and $\beta = 0.5$. Confirming the results in [144], SVGD outperforms MAP for all learning schemes. Moreover, WFEM-GP is observed to outperform GP, PACOH-GP with $(1-\beta)N$ target tasks and partially bridge the gap to the ideal PACOH-GP with N target tasks.