Bayesian Optimization Across the Spectrum of Knowledge

Bayesian Optimization Across the Spectrum of Knowledge

Enhancing Efficiency through Beliefs, Information and Assumptions

by Carl Hvarfner

LUND
UNIVERSITY

Thesis for the degree of Doctor of Philosophy
Supervisor: Associate Professor Luigi Nardi
Co-supervisor: Professor Jacek Malec
Faculty opponent: Associate Professor Luigi Acerbi

Public Defense: March 17, 2024, at 13:00 in M:D, M-building, LTH, Ole Römers väg, 223 63 Lund, Sweden.

Organization
LUND UNIVERSITY

Department of Computer Science
Box 118
Klas Anshelms väg 10–223 63 LUND
Sweden

Author(s)
Carl Hvarfner

Document name
DOCTORAL DISSERTATION

Date of disputation
2025-03-17

Sponsoring organization
The Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

Title and subtitle
Bayesian Optimization Across the Spectrum of Knowledge: Enhancing Efficiency through Beliefs, Information and Assumptions

Abstract
Bayesian Optimization has emerged as a crucial technique for optimizing costly, black-box functions where each evaluation comes at a high cost, such as in scientific experiments, and machine learning hyperparameter optimization. By combining probabilistic modeling with sequential decision-making, Bayesian Optimization achieves efficient exploration, guiding the search toward optimal parameters with minimal data. However, real-world applications present three main challenges: leveraging expert knowledge, ensuring accurate model assumptions, and managing high-dimensional search spaces.

This thesis addresses these challenges by advancing Bayesian Optimization in three key areas. First, it develops methods to incorporate practitioner insights directly into the optimization process, using domain expert knowledge to guide the search more efficiently and reduce the need for extensive evaluations. Second, it proposes techniques for dynamically validating and adapting model assumptions, enabling the Gaussian Process surrogates commonly used in Bayesian Optimization to align more closely with the complexities of real-world objective functions. Finally, this work introduces adaptive strategies for high-dimensional optimization, allowing Bayesian Optimization to focus on relevant subspaces and improve sample efficiency in vast parameter spaces, thereby mitigating the "Curse of Dimensionality."

These contributions collectively enhance Bayesian Optimization's robustness, adaptability, and efficiency, positioning it as a more powerful tool for sample-efficient optimization in complex, resource-intensive scenarios. By demonstrating these improvements through theoretical insights and empirical evaluations, this thesis establishes a pathway for more effective Bayesian Optimization in diverse, real-world applications where data is sparse and costly to obtain.

Key words Machine Learning, Artificial Intelligence, Optimization, Black-box optimization, Gaussian process, Active Learning, Bayesian Experimental design

Classification system and/or index terms (if any)

Supplementary bibliographical information

Language
English

ISSN and key title

ISBN
978-91-8104-388-4 (print)
978-91-8104-389-1 (pdf)

Recipient's notes

Number of pages
250

Price

Security classification

Bayesian Optimization Across the Spectrum of Knowledge

Enhancing Efficiency through Beliefs, Information and Assumptions

by Carl Hvarfner

LUND
UNIVERSITY

Funding information: This thesis was funded by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation..

Faculty of Engineering, Department of Computer Science

ISBN: 978-91-8104-388-4 (print)
ISBN: 978-91-8104-389-1 (pdf)
ISSN: <1404-1219>

Printed in Sweden by E-husets Tryckeri, Lund University, Lund 2024

Contents

List of publications

This thesis is based on the following publications:

I **πBO: Augmenting Acquisition Functions with User Beliefs for Bayesian Optimization**
C. Hvarfner, D. Stoll, A. Souza, M. Lindauer, F. Hutter, L. Nardi.
10th International Conference of Learning Representations (ICLR 2022).

II **Joint Entropy Search for Maximally-Informed Bayesian Optimization**
C. Hvarfner, F. Hutter, L. Nardi.
36th International Conference on Neural Information Processing Systems (NeurIPS 2022).

III **Self-Correcting Bayesian Optimization through Bayesian Active Learning**
C. Hvarfner, E. Hellsten, , F. Hutter, L. Nardi.
37th International Conference on Neural Information Processing Systems (NeurIPS 2023).

IV **A General Framework for User-Guided Bayesian Optimization**
C. Hvarfner, F. Hutter, L. Nardi.
12th International Conference of Learning Representations (ICLR 2024).

V **Leveraging Axis-Aligned Subspaces for High-dimensional Bayesian Optimization with Group Testing**
E. Hellsten*, **C. Hvarfner***, L. Papenmeier*, L. Nardi.
Preprint.

VI **Vanilla Bayesian Optimization Performs Great in High Dimensions**
C. Hvarfner, E. Hellsten, L. Nardi.
40th International Conference on Machine Learning (ICML 2024).

Publications not included in this thesis:

VII **Learning Skill-Based Industrial Robot Tasks with User Priors**

M. Mayr, **C. Hvarfner**, K. Chatzilygeroudis, L. Nardi, V. Krueger.
2022 IEEE 18th International Conference on Automation Science and Engineering (CASE 2022).

VIII **PriorBand: Practical Hyperparameter Optimization in the Age of Deep Learning**

N. Mallik, E. Bergman, **C. Hvarfner**, D. Stoll, M. Janowski, M. Lindauer, L. Nardi, F. Hutter.
37th International Conference on Neural Information Processing Systems (NeurIPS 2023).

IX **CATBench: A Compiler Autotuning Benchmarking Suite for Black-box Optimization**

J. Tørring*, **C. Hvarfner***, L. Nardi, M. Själander.
Preprint, 2024.

Acknowledgements

I would like to express my heartfelt gratitude to my main supervisor, Luigi Nardi. I am deeply thankful to Luigi for taking a chance on a candidate with no significant prior research experience, questionable coding skills, and no familiarity with the academic world. I appreciate the high standards you set—standards I wasn't even aware existed - for what was expected in our research, and I gave my utmost effort to meet them.

I extend my thanks to Jacek for his unwavering support and remarkable patience, especially when I approached tasks in unconventional ways. I know I created more work for you than I should have, and I'm truly grateful for your understanding. For the same reason, I would like to thank Elin Anna Topp for her support.

A special thanks to my colleagues Leonard Papenmeier, Kenan Sehic and Erik Hellsten. Additional thanks to Erik for letting me stay with you, for enduring our debates, and for being an incredible collaborator. Your willingness to help, your thoughtful opposing views (less intrusive than mine), and our shared groove often led to some of the best research outcomes. Moreover, I would also like to thank Artur Souza for being an incredibly helpful and patient co-author on my very first paper - it most certainly would not have happened without you.

I am also immensely grateful to Frank Hutter for many exciting collaborations and for being so enthusiastically involved. Your informed opinions and constructive critiques, often delivered within minutes of hearing my ideas, made working with you both productive and enjoyable. I am equally thankful to Marius Lindauer, who, like Frank, was always eager to engage and made sure to include me in events and discussions within your research groups.

I would also like to thank the many researchers who have shown interest in my work, offered valuable critiques, and engaged in fascinating, fruitful discussions. Henry, Danny, Eddie, and James come to mind, but I know there are many more of you—thank you all! My appreciation extends to my colleagues at Meta for a memorable six months: to Eytan for your enthusiasm in exploring half-baked ideas, to Max for your structured approach and puns, and to David for sharing my perspectives on what constitutes good (and not-so-good) research.

Finally, I owe a great deal of gratitude to Johan for convincing me that pursuing a Ph.D. was even an option, and to my parents for instilling in me the confidence to believe that no intellectual challenge is ever too great.

Carl Hvarfner
Lund 2024

Popular summary in English

Trial and error is a fundamental approach to problem-solving, evident in both everyday life and industrial settings. When faced with an unknown, we often begin by experimenting with different options, observing what works best, and gradually refining our choices. For example, a chef might adjust the ingredients in a recipe based on taste, or an engineer may test various parameters of a machine to optimize its performance. Over time, this process becomes more informed as we internalize what each adjustment achieves, allowing us to make smarter choices with each iteration.

In industrial contexts, trial and error is particularly valuable but comes at a high cost. Adjusting manufacturing parameters, for example, can improve product quality, yet each test may consume significant time, materials, and labor. Unlike a personal or small-scale trial-and-error process, the adjustments required in an industrial setting are often too complex to intuitively internalize, making it difficult to predict which changes will yield the best results. Thus, effective trial and error in these contexts requires a systematic approach that learns from each past attempt to guide future decisions, focusing resources on promising adjustments while steering clear of less productive options. The ultimate goal is to automate this learning process - to internalize it, algorithmically.

Bayesian Optimization (BO) is designed to tackle these challenges by finding optimal solutions with minimal trials, using past observations to help inform where to explore next. BO relies on a surrogate model that learns from prior evaluations to estimate the most promising areas for subsequent testing, enabling a systematic and efficient search for the best solution, whether in optimizing a recipe or fine-tuning an industrial process.

However, traditional BO methods have limitations in that they primarily learn from observed data while often overlooking insights from experts or other sources of information. Practitioners frequently bring valuable knowledge to the process, such as experience-based insights into likely successful parameter ranges or system behaviors. Despite this, conventional BO frameworks are typically unable to incorporate such prior knowledge, instead assuming that each scenario lacks meaningful initial understanding of the objective function landscape. This thesis addresses this limitation by developing methods that directly integrate practitioner beliefs, prior information, and foundational assumptions into the BO framework, thereby enhancing both its efficiency and effectiveness.

Populärvetenskaplig Sammanfattning på Svenska

Att empiriskt testa sig fram (eng. "trial-and-error") är en grundläggande metod för problemlösning, som uppkommer både i vardagen och i industriella sammanhang. När vi ställs inför ett okänt problem börjar vi ofta med att experimentera med olika alternativ, observerar vad som fungerar bäst och förfinar gradvis våra val därefter. Exempelvis kan en kock justera ingredienserna i ett recept utifrån önskad smak, eller en ingenjör kan testa olika inställningar på en maskin för att optimera dess prestanda. Med tiden blir båda dessa processer mer informerade allt eftersom vi internaliserar vad varje justering åstadkommer, vilket gör att vi kan fatta smartare beslut med varje nytt test.

I industriella sammanhang är trial-and-error särskilt relevant, men förknippat med höga kostnader. Att justera parametrar i en tillverkningsprocess kan förbättra produktens kvalitet, men kan kräva betydande resurser i form av tid, material och arbetskraft. Dessutom är de justeringar som krävs i industrin ofta för komplexa för att internaliseras, vilket gör det svårt att intuitivt och pålitligt prediktera vilka förändringar som ger bäst resultat. Effektiv trial-and-error i dessa sammanhang kräver därför en systematisk metod som lär av tidigare försök - som styr framtida beslut med hjälp av en begränsad mängd historisk data. Resurser bör fokuseras på lovande konfigurationer medan mindre produktiva alternativ undviks, med det övergripande målet att automatisera inlärningsprocessen

Bayesiansk optimering (BO) är ett verktyg för att hantera dessa utmaning - att hitta högpresterande lösningar med minimalt antal försök. Genom att använda tidigare data informerar BO beslut om hur framtida tester ska ske. BO bygger på en modell som emulerar målfunktionen och lär sig från tidigare data för att uppskatta de mest lovande områdena för kommande tester. Detta möjliggör en systematisk och effektiv sökning efter högkvalitativa lösningar, oavsett om målfunktionen är att optimera ett recept eller kalibrera en industriell process.

Dock har traditionella metoder inom BO begränsningar. Eftersom de främst lär sig av observerade data, förbiser de insikter från andra informationskällor. Utövare och ämnesexperter bidrar ofta med värdefull kunskap till processen, såsom erfarenhetsbaserade insikter om högpresterande testparametrar eller systembeteenden. Trots detta saknar konventionella BO-ramverk förmågan att integrera sådan förkunskap, och antar istället att man i varje ny situation saknar en meningsfull initial förståelse av målfunktionens utformning och beteende. Denna avhandling adresserar denna begränsning genom att utveckla metoder som direkt integrerar användares förståelse, tidigare information och antaganden i BO-ramverket, vilket förbättrar både dess effektivitet och funktionalitet.

Bayesian Optimization across the Spectrum of Knowledge

Enhancing Efficiency through Beliefs, Information and Assumptions

1 Motivation and Objectives

1.1 Motivation

Sample-efficient optimization is crucial in applications where each experiment or trial comes with a high cost. Consider optimizing the fuel efficiency of an industrial gas turbine [133], a process that depends on precise settings of air-to-fuel ratios, combustion temperature, and compressor pressure. Engineers aim to find ideal parameters to maximize efficiency while minimizing emissions, but evaluating each potential setting involves running the turbine, consuming fuel, and incurring wear on the equipment. Each trial requires hours of operation and significant fuel costs, making it impractical to test parameters exhaustively.

In this scenario, the objective function — fuel efficiency — behaves like a black box, revealing possibly noise-distorted outcomes only after each run, without providing insights into gradients or precise parameter interactions. Moreover, the measurements are likely to be imperfect, and corrupted by noise. This setup calls for zeroth-order optimization, where decisions must be made based solely on the observed outcomes of previous trials. Because each test is costly and time-consuming, the optimization process must be sample-efficient, minimizing the number of evaluations.

In the quest for optimizing these complex systems efficiently, Bayesian optimization (BO) [104, 147, 43, 49] has emerged as a powerful paradigm, achieving unparalleled sample efficiency in these contexts [35, 51, 102, 18]. Rooted in

Bayesian inference and sequential decision-making, Bayesian optimization offers a principled framework for tackling optimization tasks where evaluating the objective function is costly or impractical. From tuning hyperparameters of machine learning models [**?** 136] to optimizing experimental parameters in scientific research [73, 48], the relevance of Bayesian optimization pervades numerous domains. Its ability to guide the search process intelligently, leveraging past observations to inform future decisions, not only accelerates the convergence to optimal solutions but also facilitates robustness against noise and uncertainty.

In the landscape of optimization methodologies, BO, coupled with Gaussian Processes (GPs) [143, 132], stands out particularly in scenarios characterized by limited data availability, commonly referred to as the *small data* regime. Unlike conventional optimization techniques tailored for *big data* settings, where vast amounts of data enable statistical inference and model training at scale [83, 164], BO thrives in situations where data points are sparse and expensive to obtain. GPs, serving as probabilistic surrogate models in BO, offer a flexible framework for capturing uncertainty and modeling complex, nonlinear relationships with minimal data requirements. In contrast to deep learning models [86], which often demand large amounts of labeled data for effective training[83, 164], GPs excel in interpolating from a modest number of observations, making them inherently well-suited for small data regimes. Moreover, the Bayesian framework provides a principled approach to incorporating prior knowledge and beliefs about the optimization problem, further enhancing the efficiency and effectiveness of the optimization process in data-limited scenarios.

However, it is important to distinguish between the types of priors that GPs can naturally incorporate and the broader forms of knowledge that practitioners often possess. GPs already allow the injection of priors in the form of assumptions about smoothness, noise levels, and covariance structure through the kernel function and its hyperparameters [132, 32, 37]. These priors reflect structural properties of the objective function, such as expected continuity or periodicity, and guide the surrogate model accordingly. By contrast, the multiple types of user knowledge explored in this thesis extend beyond these existing GP priors. Practitioners often have intuitive or empirical beliefs [16] about specific aspects of the optimization problem, such as the likely region of the input space where the optimum is located [153], an upper bound on achievable outcomes, or preference relations between certain parameters. These are higher-level, application-specific insights that are orthogonal to the structural priors traditionally captured by GPs.

For example, a user might suspect, based on domain expertise, that the optimal learning rate for a neural network lies within a specific range [166, 145, 127]

or that a certain combination of hyperparameters is unlikely to yield desirable results. Such beliefs can be framed as priors over the location of the optimum or the optimal value, or even as relational preferences among input-output pairs. These types of priors, which are informed by practical experience and context, have the potential to accelerate convergence and reduce the number of evaluations required. However, conventional BO frameworks lack mechanisms to systematically incorporate these forms of knowledge, leading to suboptimal utilization of the information available.

Thus, *the first research question addresses how to effectively integrate intuitive knowledge from users to improve BO's time-to-performance.* To address these challenges, this thesis explores methods to effectively integrate practitioners' intuitive knowledge into BO, bridging the gap between user-defined priors and the traditional GP framework. By enabling BO to systematically leverage these insights, the goal is to enhance its sample efficiency and improve time-to-performance. This distinction between structural priors inherent to GPs and user-defined beliefs highlights the unique focus of this work on expanding the flexibility and utility of BO in real-world applications.

The second research question is motivated by the need to validate and exploit model-level assumptions that underlie the BO process. In BO, GP models are typically used to approximate the objective function. The success of these models depends not only on the data collected but also on assumptions about the model's structure, such as smoothness, noise level, and function shape [32, 132]. While the primary objective of BO is to locate the optimum, acquiring auxiliary information during the optimization process — such as insights into model hyperparameters, the optimal value of the objective or the appropriateness of structural assumptions — can significantly enhance this search. Such auxiliary information provides additional context, enabling the algorithm to refine its surrogate model, validate critical assumptions, or adapt to unknown complexities in the objective function.

For example, accurately learning the hyperparameters of the Gaussian Process model calibrates the uncertainty estimates in the posterior, leading to more informed acquisition decisions. Similarly, identifying when assumptions about the function's smoothness or noisiness are misaligned with reality can help the algorithm recalibrate its approach, improving sample efficiency. Therefore, the second research question asks: how can BO best utilize auxiliary information about model-level assumptions to improve optimization efficiency while still maintaining focus on the primary objective of locating the optimum?

The third research question is driven by the demands of high-dimensional opti-

mization tasks, where the "curse of dimensionality" creates substantial obstacles. In high-dimensional spaces, conventional BO methods often perform poorly due to sparse data coverage and increased model complexity, making it challenging to identify relevant patterns in the objective function. To address this, simplifying assumptions—such as focusing on a lower-dimensional active subspace [167, 125, 115] - are frequently introduced to make the optimization problem more tractable. Importantly, these assumptions are not necessarily made because they reflect true beliefs about the structure of the objective function, but rather because high-dimensional problems are otherwise infeasible to optimize. These assumptions are imposed to facilitate optimization by ensuring that the problem can be effectively modeled and solved within the constraints of the BO framework.

For instance, active subspace methods assume that the objective depends on only a subset of dimensions, but such assumptions often remain unverified and may misrepresent the actual structure of the problem. Developing strategies that either validate or bypass these assumptions, depending on the context, could greatly improve BO's ability to handle high-dimensional tasks. Therefore, the third research question seeks to understand how assumptions imposed on the surrogate model can be leveraged or adapted to enhance high-dimensional BO's efficiency, where direct optimization without simplifying assumptions is impractical.

These three areas — incorporating user knowledge, validating model assumptions, and navigating high-dimensional spaces — each represent a critical and unique challenge in the field of BO. Addressing these challenges not only pushes the boundaries of sample-efficient optimization, but expands BO's applicability across complex, real-world scenarios where data is limited and evaluations are costly.

1.2 Research Objectives

The objective of this thesis is to advance Bayesian optimization methods by incorporating auxiliary information to enhance its efficiency, either in terms of accelerated time-to-performance or quality of the terminal solution found, as measured on the objective function. Additional information may take various forms, such as insights provided by practitioners, default assumptions tailored to specific problem contexts, or data deduced dynamically throughout the optimization process, that does not directly tie into the task of optimizing the objective. By *improve*, we specifically mean achieving greater sample efficiency — reducing the number of evaluations needed to reach a desirable level of performance — and, ultimately, finding better parameters and outcomes for a

given objective. This thesis aims to demonstrate how these auxiliary-informed methods can outperform traditional Bayesian optimization by capitalizing on previously underutilized information sources. In doing so, this thesis pushes the boundaries of sample-efficient optimization, demonstrating new pathways for achieving faster, more accurate solutions in challenging problem spaces. With this in mind, the objective of enhancing Bayesian optimization through the consideration of auxiliary information and objectives is subdivided into three distinct research questions.

1.3 Research Questions

Given the motivation outlined in Sec. 1.1, three primary research questions are identified:

RQ1. How can practitioners' intuitive knowledge and empirical beliefs about the objective function be systematically integrated into Bayesian optimization to accelerate convergence and improve time-to-performance?

RQ2. How can auxiliary information about model hyperparameters or structural assumptions, be dynamically validated and exploited during the optimization process to improve the efficiency and reliability of Bayesian optimization?

RQ3. How can simplifying assumptions imposed on the surrogate model be leveraged to facilitate efficient Bayesian optimization in high-dimensional problem settings, where direct optimization is otherwise impractical?

1.4 Thesis Outline

The thesis is a compilation in which the first part serves as the *Kappa*, providing an introduction to the research field and outlining the scope of the work conducted in this thesis. The second part contain the individual papers that comprise the thesis. The thesis is structured as follows:

- Section 1 motivates the topic of research, outlines the overarching objective of the thesis, and presents three concrete research questions (RQs).
- Section 2 provides foundational material on probabilistic modeling, setting the stage for the subsequent exploration of GP and BO.

- Section 3 delves into Gaussian processes, explaining their role as priors over functions, their formulation, and their practical implementation for learning and hyperparameter optimization.
- Section 4 discusses Bayesian Optimal Experimental Design, highlighting its relevance to Bayesian optimization and exploring the parameters of interest, its relevance in Gaussian process hyperparameterization and practical considerations in its implementation.
- Section 5 focuses on Bayesian optimization, detailing surrogate modeling approaches, acquisition functions, and the iterative nature of the Bayesian optimization process.
- Section 6 summarizes the key contributions of the thesis, organizing them by the three research questions: user-guided Bayesian optimization, leveraging auxiliary model-level objectives, and high-dimensional Bayesian optimization.
- Section 7 concludes the thesis with final reflections, highlighting the implications of the research, its limitations, and avenues for future work.

The second part of the thesis consists of six individual papers that represent the core of the research contributions. Each paper is included in its entirety, offering comprehensive details on the methodologies, experiments, results, and discussions that align with the research objectives outlined in the first part of the thesis.

- Paper I introduces πBO, a user-guided acquisition function that incorporates practitioner beliefs about the location of the optimum into the optimization process. The method demonstrates improved time-to-performance in hyperparameter optimization (HPO) tasks and provides theoretical convergence guarantees, addressing RQ1.
- Paper II presents Joint Entropy Search (JES), an information-theoretic acquisition function that reduces entropy over joint input-output spaces. JES achieves strong performance when the surrogate model is accurate, addressing RQ2.
- Paper III introduces Self-Correcting Bayesian Optimization (SCoreBO), which integrates active learning into BO to dynamically refine model hyperparameters during optimization. It enhances model reliability and improves optimization efficiency under model-level uncertainty, addressing RQ2.

- Paper IV proposes Collaborative Bayesian Optimization (ColaBO), a framework that incorporates diverse user-defined priors, such as preferences and bounds, into the surrogate model. This method improves optimization efficiency while increasing user control, addressing RQ1.
- Paper V introduces Group Testing Bayesian Optimization (GTBO), a high-dimensional BO method that uses adaptive group testing to identify relevant dimensions. This two-step approach combines feature selection with efficient low-dimensional optimization, addressing RQ3.
- Paper VI presents a plug-and-play prior adjustment for Gaussian process kernels, enabling efficient high-dimensional optimization without complex structural assumptions. This simple approach demonstrates strong performance across a range of challenging tasks, addressing RQ3.

2 Probabilistic Modeling

Probabilistic modeling [110] provides a framework for capturing uncertainty in predictions by representing quantities of interest as random variables with associated probability distributions. Instead of yielding a single, fixed outcome, probabilistic models express uncertainty about outcomes, allowing predictions that reflect the variability inherent in complex data. In probabilistic modeling, we aim to find a model $p(y|x)$ that describes the likelihood of observing y given input x, where y represents an outcome we want to predict.

Bayesian modeling extends this framework by incorporating prior beliefs about model parameters before observing any data. These prior beliefs, represented by a prior distribution $p(\xi)$ over the parameters ξ are updated in light of new data through Bayes' theorem. When given observed data $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the goal is to compute the posterior distribution $p(\xi|D)$, which represents our updated belief about ξ after accounting for the evidence. Mathematically, Bayes' theorem defines the posterior as:

$$p(\xi|D) = \frac{p(D|\xi)p(\xi)}{p(D)}, \tag{1}$$

where the

- **Prior** $p(\xi)$ represents our initial beliefs about the parameters ξ before seeing any data,
- **Likelihood** $p(D|\xi)$ represents the probability of observing the data D given the parameters ξ, often derived from a probabilistic model $p(y|\boldsymbol{x}, \xi)$ applied across observed points,
- **Posterior** $p(\xi|D)$ is the updated distribution of ξ after observing D, which combines prior beliefs with information from the data.
- **Evidence** $p(D)$: is a normalization factor ensuring that $p(\xi|D)$ integrates to 1, computed as $p(D) = \int p(D|\xi)p(\xi)\, d\xi$.

Bayesian modeling offers several advantages: the posterior $p(\xi|D)$ not only provides a point estimate of parameters but also quantifies the uncertainty around these estimates. The ability to quantify uncertainty in its predictions makes Bayesian modeling particularly useful in real-world applications, where decision-making generally carries risk and environments are generally uncertain.

3 Gaussian Processes

Gaussian processes (GPs) [143, 132] provide a flexible and powerful framework for modeling a distribution over an unknown function [172], particularly in cases where the underlying function is expensive to evaluate or difficult to model analytically. GPs are a class of non-parametric models, meaning they can model functions of varying complexity without relying on a fixed number of parameters. This makes GPs especially useful in Bayesian optimization, where they serve as surrogate models that approximate the unknown objective function based on observed data. By defining a prior over functions, GPs make probabilistic predictions, capturing both the mean and variance of the function at unobserved locations.

3.1 Kernel Methods

Kernel methods [141, 146, 61] form the mathematical backbone of GPs by specifying how inputs are related, allowing GPs to make smooth, correlated predictions across the input space. A kernel $k(x, x')$, also known as a covariance function, measures the similarity between two inputs x and x'. This similarity, defined through the kernel, determines how changes at one point influence values at other points, effectively controlling the function's smoothness, periodicity, and other structural properties.

3.1.1 General Form of Kernels

A kernel is a symmetric, positive semi-definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which computes a similarity measure between two D-dimensional inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$, where $\boldsymbol{x} = [x_1, x_2, \ldots, x_D]$ [132, 32] in the domain $\mathcal{X}$. One of the simplest and most commonly used kernels is the squared exponential (or radial basis function, RBF) kernel:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\ell^2}\right) \tag{2}$$

which encodes assumptions about the function's smoothness and correlation. where σ_f^2 is the variance (controlling the overall magnitude of the function) and ℓ is the length-scale parameter (controlling the smoothness of the function). The Squared Exponential, or RBF (Radial Basis Function) kernel assumes that

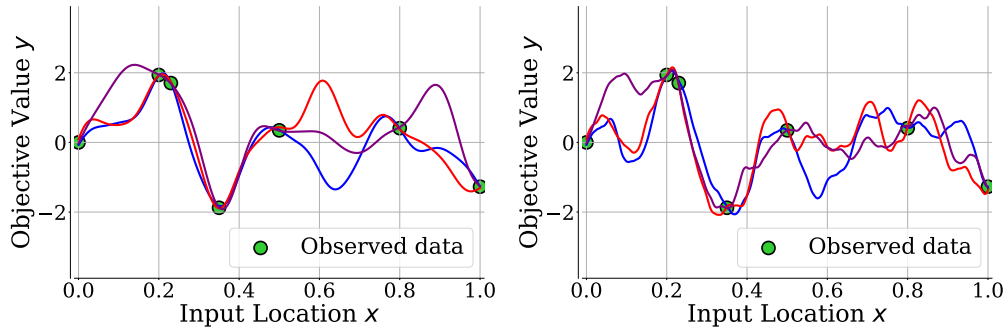

Fig. 1: Visual difference between an RBF and Matern kernel. **a)** Three sample functions (red, blue, purple lines) drawn from a GP with an RBF Kernel. **b)** Sample functions drawn from a GP with a Matern($\nu = 5/2$) Kernel. Functions drawn from a Matern kernel exhibit a lower degree of smoothness, as the samples are visually more jagged than those drawn from the RBF kernel.

the function is infinitely differentiable, making it suitable for modeling smooth functions.

Other common kernels include the Matérn [100] kernel,

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\boldsymbol{x} - \boldsymbol{x}'\|}{\ell} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}\|\boldsymbol{x} - \boldsymbol{x}'\|}{\ell} \right), \tag{3}$$

where ν controls the degree of smoothness of f, is the modified Bessel function, and Γ is the Gamma function. Thus, the Matérn kernel can capture functions that are more or less jagged, converging to the RBF kernel in the limit of $\nu = \infty$. Lastly, there exist more exotic kernels, such as the periodic kernel,

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{2\sin^2\left(\frac{\pi|\boldsymbol{x}-\boldsymbol{x}'|}{p}\right)}{\ell^2} \right), \tag{4}$$

which is useful for modeling periodic functions, and the spectral mixture kernel [173], which is useful for extrapolation.

3.1.2 Properties of Kernels

Each kernel has distinct properties that make it suitable for different types of functions. Kernels can be combined [33, 32, 95] (e.g., summed or multiplied) to create composite kernels that capture complex relationships in the objective,

allowing the GP to model functions with multiple attributes, such as smooth trends with periodic fluctuations. The choice of kernel is critical, as it defines the structure of the GP prior and significantly impacts the GP's ability to fit observed data and make accurate predictions in unobserved regions. The kernel function, $k(x, x')$ plays a central role in shaping the GP prior and influences predictions on unobserved points based on observed data points.

3.1.3 Automatic Relevance Determination

Automatic Relevance Determination (ARD) is an extension of standard kernel functions in GPs that enables the model to identify the impact of each input dimension independently [172, 132]. With ARD, each input dimension is assigned its own length-scale parameter, allowing the GP to vary the degree of influence each dimension has on the predictions. For example, The ARD version of the RBF kernel is given by:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{(x_d - x_d')^2}{\ell_d^2}\right), \tag{5}$$

where D is the dimensionality of the input space, ℓ_d represents the length-scale for the d-th dimension, and σ_f^2 is the signal variance.

The length-scale ℓ_d in ARD controls how much influence the d-th dimension has on the function's output. A large ℓ_d suggests that the function varies slowly along that dimension, indicating that the dimension is less relevant to the model. Conversely, a small ℓ_d implies that the function is more sensitive to variations in that dimension, making it more influential in the model's predictions. This mechanism allows GPs to automatically adjust the importance of each input dimension, improving model flexibility. Moreover, the ARD framework provides a way to interpret the influence of each feature, offering insights into which dimensions meaningfully impact the predictive performance.

3.2 Updating the GP with Data

When observations are available, GPs update their beliefs by conditioning on this data to form a posterior distribution. Suppose we have observed n data points, $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $y_i = f(\boldsymbol{x}_i) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$ representing Gaussian noise. The GP posterior distribution for a new input $\boldsymbol{x}_*$ is then computed based on both the prior and the likelihood of the data, yielding the posterior mean

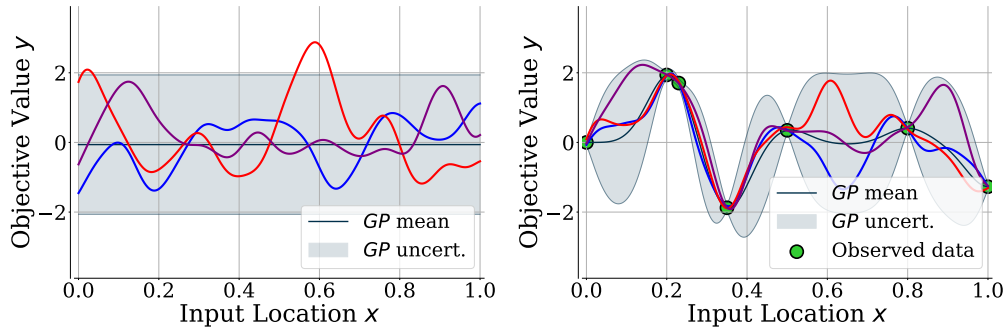

Fig. 2: A Gaussian Process updating with data. **a)** Three samples (red, blue, purple lines) from a GP with an RBF Kernel, conditioned on no data. Predictive mean is visualized as a dark grey line. Predictive uncertainty is visualized as a light grey band. **a)** Predictive moments (mean, variance) are updates according to Eq. (6) and Eq. (7), respectively, and three samples from the resulting GP posterior are drawn. The space of plausible functions has shrunk as a result of the acquisition of data.

$\mu(\boldsymbol{x}_*)$ and and variance $\sigma^2(\boldsymbol{x}_*)$ at $\boldsymbol{x}_*$ as [132]:

$$\mu(\boldsymbol{x}_*) = k(\boldsymbol{x}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}\mathbf{y}, \tag{6}$$

$$\sigma^2(\boldsymbol{x}_*) = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - k(\boldsymbol{x}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}k(\mathbf{X}, \boldsymbol{x}_*), \tag{7}$$

where

- $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top$ represents the observed input points,
- $\mathbf{y} = [y_1, \ldots, y_n]^\top$ represents the observed outputs,
- $K(\mathbf{X}, \mathbf{X})$ is the covariance matrix of the observed inputs,
- $k(\boldsymbol{x}_*, \mathbf{X})$ is the covariance vector between $\boldsymbol{x}_*$ and the observed inputs X,
- I is the identity matrix.

The GP posterior mean, $\mu(\boldsymbol{x}_*)$ provides a point estimate of $f(\boldsymbol{x}_*)$, while $\sigma^2(\boldsymbol{x}_*)$ gives an estimate of the uncertainty around $\mu(\boldsymbol{x}_*)$.

3.3 A Prior over Functions

A Gaussian process (GP) defines a prior over functions, making it a powerful tool for modeling unknown functions in a Bayesian framework. Formally, a

GP is a collection of random variables, any finite subset of which follows a multivariate Gaussian distribution [132]. This property allows a GP to represent distributions over functions in a way that naturally incorporates uncertainty about the function's values at unobserved points.

A Gaussian process for a function $f(\boldsymbol{x})$ is denoted as:

$$f(\boldsymbol{x}) \sim \mathcal{GP}\left(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\right), \tag{8}$$

where where $m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})]$ is the mean function and $k(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{Cov}(f(\boldsymbol{x}), f(\boldsymbol{x}'))$ is the covariance function., which defines the similarity between function values at points $\boldsymbol{x}$ and $\boldsymbol{x}'$.

The GP prior thus represents an infinite-dimensional distribution over functions. Any set of points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$ corresponds to a multivariate Gaussian distribution over the associated function values $f(X) = [f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_M)]^\top$, where

$$f(X) \sim \mathcal{N}(m(\mathbf{X}), K(\mathbf{X}, \mathbf{X})), \tag{9}$$

with $m(X) = [m(\boldsymbol{x}_1), m(\boldsymbol{x}_2), \ldots, m(\boldsymbol{x}_n)]^\top$ denoting the mean vector and $K(X, X)$ denoting the covariance matrix where $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)\ \forall i, j \in [1, M]$. The multivariate Gaussian property allows the GP to model function values at observed points as well as make predictions at unobserved points, since the joint distribution of $f(X)$ and $f(\boldsymbol{x}_*)$ remains Gaussian:

$$\begin{bmatrix} f(\mathbf{X}) \\ f(\boldsymbol{x}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{X}) \\ m(\boldsymbol{x}) \end{bmatrix}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) & k(\mathbf{X}, \boldsymbol{x}) \\ k(\boldsymbol{x}, \mathbf{X}) & k(\boldsymbol{x}, \boldsymbol{x}) \end{bmatrix}\right). \tag{10}$$

3.3.1 A Multitude of Alternative Priors

While a Gaussian Process (GP) provides a prior over functions, it is important to clarify what this entails and, equally, what it does not. A GP is a prior over the values of a function $f(\boldsymbol{x})$ defining the distribution of possible functions that could describe the relationship between inputs $\boldsymbol{x}$ and outputs $f(\boldsymbol{x})$. Through the mean function $m(\boldsymbol{x})$ and the covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$, a GP encodes structural assumptions about the function, such as smoothness, periodicity, or expected variance. For instance Fig. 1, displays how the choice of kernel k dictates how rapidly the function can change over the input space.

However, while GPs are a prior over functions, they are not inherently priors over specific properties of the function, such as the location of the optimum [65] $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$ within a bounded domain $\mathcal{X}$, the optimal value within

the same domain $f^* = f(\boldsymbol{x}^*)$, or relational preferences $f(\boldsymbol{x}_i) \geqslant f(\boldsymbol{x}_j)$ among inputs $\boldsymbol{x}_i, \boldsymbol{x}_j$. These higher-level properties often arise from application-specific requirements or practitioner insights, which are external to the GP's intrinsic representation. Moreover, these insights are arguably simpler to obtain and reason about than properties of the kernel function, as these may be unintuitive to non-GP experts [153]. Such attempts to encode such properties into the GP have been explored by [153, 34] in the case of a distribution over the optimal input, and [69, 135, 120] in the case of optimal value. Paper I proposes an approach to consider distributions over the optimal input location in a BO context, whereas Paper IV explores a general approach to encoding all the aforementioned properties into the GP in a principled Bayesian manner, thereby extending the GP's ability to encode beliefs held by practitioners.

3.3.2 Sampling from the GP

To illustrate the concept of a GP as a prior over functions, we can sample functions from the prior distribution defined by $m(\boldsymbol{x})$ and $k(\boldsymbol{x}, \boldsymbol{x}')$. For a dense set of input points, we draw samples from the multivariate Gaussian in Eq. (10). Each sample represents a possible realization of the function $f(\boldsymbol{x})$ might look like. Thus, these samples reflect the GP's belief about what the function might look like before observing any data, as well as after. By observing data, the plausible set of functions shrinks, as functions that do not accurately interpolate the data become increasingly less probable.

For a finite set of k query locations ($\mathbf{X} = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$), samples can be generated using the classical location-scale transformation of Gaussian random variables, $f(\mathbf{X}) = \mu_n(\mathbf{X}) + L\boldsymbol{\varepsilon}$, where L is the Cholesky decomposition of K and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, I)$. However, this approach is inherently computationally intensive, as it incurs a cost of $\mathcal{O}(k^3)$ due to the matrix decomposition required.

3.3.3 Decoupled Posterior Sampling

To remedy the issue of scalability in posterior sampling, $\mathcal{O}(k)$ weight-space approximations based on Random Fourier Features (RFF) [130] obtain approximate (continuous) function draws $\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{m} \boldsymbol{w}_i^\top \phi_i(\boldsymbol{x})$, where $\phi_i(\boldsymbol{x}) = \frac{2}{\ell}(\psi_i^\top \boldsymbol{x} + b_i)$. The random variables $\boldsymbol{w} \sim \mathcal{N}(0, I)$, $b_i \sim \mathcal{U}(0, 2\pi)$, and ψ_i are sampled proportional to the spectral density of k.

While achieving scalability, the seminal RFF approach by [130] suffers from the issue of variance starvation [113, 170, 177]. As a remedy, [177] decouple the draw

of functions from the approximate posterior $p(\hat{f}|\mathcal{D})$ into a more accurate draw from the prior $p(\hat{f})$, followed by a deterministic data-dependent update:

$$(\hat{f}|\mathcal{D})(\boldsymbol{x}) \overset{d}{=} \underbrace{\hat{f}(\boldsymbol{x})}_{\text{draw from prior}} + \underbrace{k_n(\boldsymbol{x})^\top (K + \sigma_\varepsilon^2 I)^{-1}(y - \hat{f}(\boldsymbol{x}) - \boldsymbol{\varepsilon})}_{\text{deterministic update}} \tag{11}$$

Eq. 11 deviates from the distribution-first approach that is typically prevalent in GPs in favor of a variable-first approach utilizing Matheron's rule [74]. Paper IV extends on the decoupled updating approach by filtering the function draws from the prior through a user-defined belief $\pi(f)$ over properties of the function f:

$$(\hat{f}|\mathcal{D}, \pi)(\boldsymbol{x}) \overset{d}{=} \underbrace{(\hat{f}|\pi)(\boldsymbol{x})}_{\text{draw from prior}} + \underbrace{k_n(\boldsymbol{x})^\top (K_n + \sigma_\varepsilon^2 I)^{-1}(y - (\hat{f}|\pi)(\boldsymbol{x}) - \varepsilon)}_{\text{deterministic update}}, \tag{12}$$

where π may encode any of the properties mentioned in Sec 3.3.1. Fig. 3 displays the belief-weighted filtering in action: samples from the prior are drawn (top right, light blue) and weighted against the prior $\pi(x)$ (green). Sampled functions are re-sampled using rejection sampling against $\pi(x)$ to obtain belief-weighted draws from the posterior[68].

3.4 Hyperparameter Learning

The kernel hyperparameters, such as σ_n^2 and $\boldsymbol{\ell} = [\ell_1, \ell_2, \ldots, \ell_D]$ in the RBF kernel with ARD, play a crucial role in shaping the GP's behavior [172]. Consequently, it has a substantial impact on the performance and ultimate success of BO, as explored in Paper III, which accelerates the hyperparameter learning to achieve a more accurate GP, and in Paper VI, where a novel regularization scheme for the GP in high dimensions yields a stable model that is more amenable to optimization than previous alternatives.

3.4.1 Learning Hyperparameters by Maximizing the Marginal Likelihood

Hyperparameters are typically learned by maximizing the marginal likelihood of the observed data [132, 94]. The marginal likelihood $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$

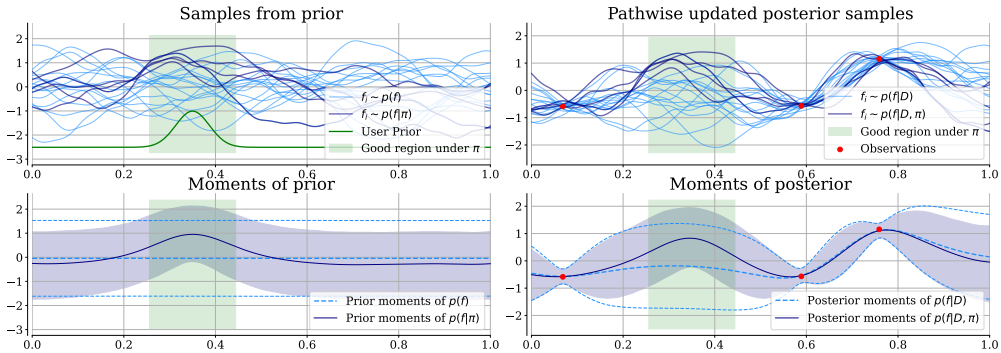

Fig. 3: (Top left) Draws from the prior $p(f)$ (light blue) and the belief-weighted prior $p(f|\pi)$ whose members are likely to have their optimum within the green region. (Top right) Decoupled updated draws based on observed data. As the green region is distant from the observed data, samples are almost unaffected by the data in this region. (Bottom left) Exact mean and standard deviation $\mu(\boldsymbol{x}), \sigma(\boldsymbol{x})$ of $p(f)$ and estimated mean and standard deviation of $p(f|\pi)$. (Bottom right) Exact $p(f|\mathcal{D})$ and estimated $p(f|\pi, \mathcal{D})$. As $p(f|\pi)$ constitutes of functions whose optimum is located within the green region the resulting model has a higher mean and lower variance within this region. Moreover, $p(f|\pi)$ globally displays lower upside variance compared to the vanilla GP.

represents the hyperparameters $\boldsymbol{\theta} = \{\boldsymbol{\ell}, \sigma_f^2, \sigma_\varepsilon^2\}$, is computed as:

$$\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^\top [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}\mathbf{y} \tag{13}$$

$$-\frac{1}{2}\log|K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I| - \frac{n}{2}\log 2\pi. \tag{14}$$

This expression balances the model fit (via the term $\mathbf{y}^\top [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}\mathbf{y}$) against model complexity (via the log-determinant term ($\log|K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I|$). The optimal model fit, the maximum likelihood estimator (MLE), under the marginal (log) likelihood criterion is obtained through maximization,

$$\boldsymbol{\theta}^*_{\text{MLE}} = \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}). \tag{15}$$

The MLE approach allows BO to adapt the GP model to observed data, improving prediction accuracy. penalizing overly complex models. In practice, the hyperparameter optimization is performed using gradient-based techniques.

3.4.2 Incorporating Hyperparameter Priors with Maximum a Posteriori Estimation

In cases where prior knowledge about the hyperparameters is available, Bayesian methods can be used to incorporate this information, leading to a more informed

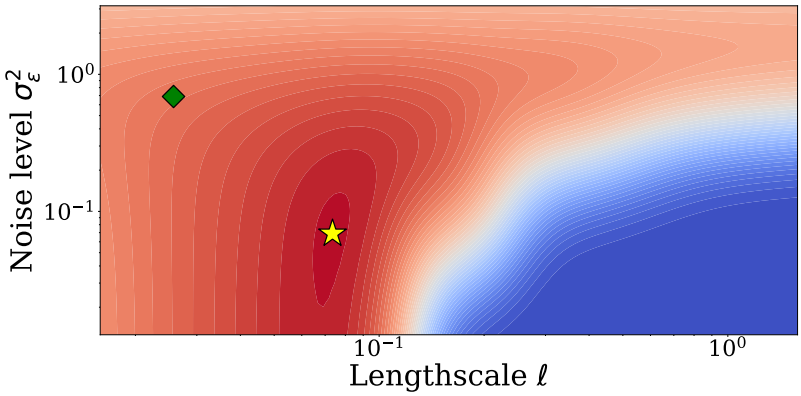

Fig. 4: Marginal Likelihood surface (Eq. (13)) for a GP with two hyperparameters: noise level σ_ε^2 and lengthscale ℓ. Red values indicate more plausible models (higher likelihood) and blue indicates less plausible models (lower likelihood). The two hyperparameters provide different explanations for the data with varying likelihoods. In Figs 5 a and b, a less plausible model (a, green square) and the most plausible model (b, yellow star) under the MLE criterion are visualized.

estimation of θ. The Maximum a Posteriori (MAP) estimation [110] approach combines this prior knowledge with the observed data by maximizing the posterior distribution over the hyperparameters:

$$p(\boldsymbol{\theta}|\mathbf{X},\mathbf{y}) \propto p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta})p(\boldsymbol{\theta}), \tag{16}$$

where $p(y|X,\theta)$ is the marginal likelihood, as described above, and $p(\theta)$ represents the prior distribution over the hyperparameters. By combining these terms, MAP estimation seeks a balance between fitting the observed data well and aligning with prior beliefs about the hyperparameters. The MAP estimate θ^*_{MAP} is subsequently obtained by maximizing the log-posterior:

$$\theta^*_{\text{MAP}} = \arg\max_{\theta} \left(\log p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\right). \tag{17}$$

For example, if we have prior knowledge that the length scale ℓ_i should be within a certain range (indicating a belief about the function's smoothness), we could place a prior on ℓ, such as a uniform prior:

$$p(\ell_i) = \mathcal{U}(a,b), \tag{18}$$

for some suitable parameters a and b, where $0 < a < b$. Typical priors for these parameters include log-normal or inverse-gamma distributions to ensure positive values for the parameter estimates.

By incorporating priors, MAP estimation provides a more robust hyperparameter estimation process, especially when data is limited or noisy. Unlike marginal

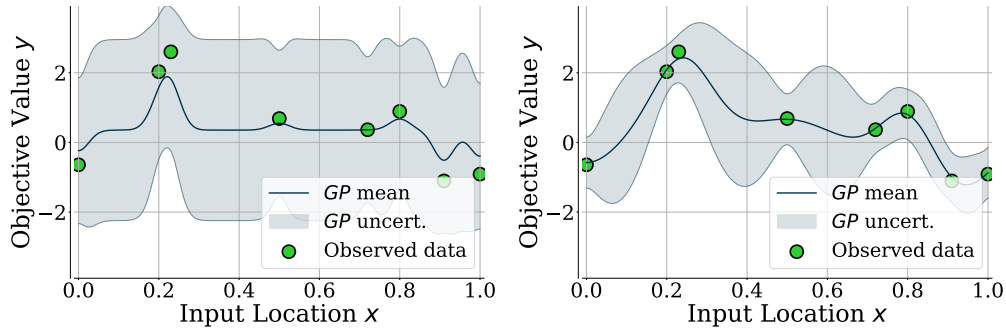

Fig. 5: Two Gaussian processes with different marginal likelihoods and a different set of observed data. **a)** A non-MLL-optimized GP under the MLE criterion, corresponding to the green square in Fig. 4. Data points close to the center are improbable, as they are either on the edge of, or outside, the confidence interval of the GP. **b)** The optimal GP posterior under the MLE criterion, corresponding to the yellow star in Fig. 4. The model constitutes the optimal trade-off between model fit and simplicity.

likelihood maximization, MAP estimation reflects a balance between observed data and prior beliefs, potentially improving predictive performance and when observations alone provide insufficient information to yield an accurate, generalizable model. Paper VI introduces a prior on the GP lengthscales that significantly enhances robustness in high-dimensional settings, enabling accurate predictions even for dimensionalities reaching into the thousands [**?**]. This prior operates under the assumption that as dimensionality D increases, the complexity of each individual dimension decreases. To formalize this, the lengthscales are scaled proportionally as $\ell_i \propto \sqrt{D}$, effectively counteracting the growing distances between points in high-dimensional spaces [82]. Specifically, this scaling is implemented by adjusting the mean μ term of a LogNormal ($\mathcal{LN}$) prior

$$\ell_i \sim \mathcal{LN}\left(\mu_0 + \frac{\log(D)}{2}, \sigma_0\right) \tag{19}$$

where (μ_0, σ_0) are parameters chosen to correspond to a one-dimensional objective. This adjustment ensures that the prior adapts naturally to increasing dimensionality while maintaining the desired properties for lower dimensions. The resulting GP hyperparameter learning is generally substantially more stable than that of MLE, and provides vastly more informative out-of-sample predictions than conventional, dimension-independent MAP [67].

3.4.3 Fully Bayesian Hyperparameter Treatment

Hyperparameter learning is central to GPs, as the choice of kernel parameters—such as length scale, signal variance, and noise level—significantly influences the model's predictions. Beyond MLE and MAP estimation, a fully Bayesian treatment [123, 112, 147, 13] provides a robust alternative by capturing uncertainty in hyperparameters explicitly, instead of optimizing them to fixed values. This approach can be particularly advantageous in BO [147, 48, 37], where accurately modeling uncertainty can enhance the sample efficiency and reliability of the GP surrogate model. Paper III exploits this fact by presenting a BO algorithm which simultaneously reduces model-level uncertainty while optimizing the objective function [66].

In a fully Bayesian approach to hyperparameter learning, we aim to capture the entire posterior distribution over the hyperparameters $\boldsymbol{\theta}$, visualized in Fig. 4, rather than finding a single, high-probability point estimate - the yellow star in Fig. 4. This is achieved by integrating over all possible values of $\boldsymbol{\theta}$ when making predictions, which provides a more nuanced representation of the uncertainty in the model. The posterior over the hyperparameters is given by Bayes' theorem:

$$p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y}|\mathbf{X})}, \tag{20}$$

where

- $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$ is the marginal likelihood, or the likelihood of observing data $\mathbf{y}$ given the inputs $\mathbf{X}$ and hyperparameters $\boldsymbol{\theta}$,
- $p(\boldsymbol{\theta})$ is the prior distribution over the hyperparameters, representing any initial beliefs,
- $p(\mathbf{y}|\mathbf{X})$ is the marginal likelihood of the data, obtained by integrating over θ:

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})\, d\boldsymbol{\theta}. \tag{21}$$

Rather than optimizing $\boldsymbol{\theta}$ directly, this approach treats $\boldsymbol{\theta}$ as a latent variable and integrates over its possible values, thus incorporating all plausible parameters of the GP's parameters into predictions.

Directly computing the posterior distribution $p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y})$ and performing the integrals for predictive inference is intractable for most non-trivial distributions. As a result, approximate inference methods are typically used. One example of

such approximate inference methods is Markov Chain Monte Carlo (MCMC) [116, 111, 59, 13]. MCMC sampling generates samples from the posterior distribution $p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y})$. Given a set of samples $\{\boldsymbol{\theta}^{(s)}\}_{s=1}^{S}$, the approximate posterior is a mixture of Gaussians with uniform mixture weights, where the mean and variance are computed as [85]

$$\mu(\boldsymbol{x}_*) \approx \frac{1}{S} \sum_{s=1}^{S} \mu(\boldsymbol{x}_*; \boldsymbol{\theta}^{(s)}), \tag{22}$$

$$\sigma^2(\boldsymbol{x}_*) \approx \frac{1}{S} \sum_{s=1}^{S} \left(\sigma^2(\boldsymbol{x}_*; \boldsymbol{\theta}^{(s)}) + (\mu(\boldsymbol{x}_*; \boldsymbol{\theta}^{(s)}) - \mu(\boldsymbol{x}_*))^2 \right) \tag{23}$$

where $(\mu(\boldsymbol{x}_*; \boldsymbol{\theta}^{(s)})$ and $\sigma^2(\boldsymbol{x}_*; \boldsymbol{\theta}^{(s)})$ denote the posterior moments of the GP with hyperparameters $\boldsymbol{\theta}^{(s)})$. MCMC provides accurate samples from the posterior, but it can be computationally demanding, especially for high-dimensional hyperparameter spaces. As alternatives, one can resort to Variational Inference (VI) [26, 163] which replaces the true posterior $p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y})$ with a simpler, tractable distribution $q(\boldsymbol{\theta})$ by minimizing the Kullback-Leibler (KL) divergence between $q(\boldsymbol{\theta})$ and the true posterior. While generally less computationally expensive, VI restricts the form of the posterior, thereby foregoing asymptotic exactness.

4 Bayesian Optimal Experimental Design

Bayesian Optimal Experimental Design (BOED) [19, 139, 131] provides a probabilistic framework for sequential learning of the parameters of a model or process through active sampling of new data. This approach is particularly valuable when data collection is expensive or time-consuming, as it prioritizes observations that will have the highest impact on reducing uncertainty about quantities of interest. Observations are acquired sequentially, with each observation providing insight that refines the model's predictions or estimates. The methodology leverages Bayes' theorem to incorporate both prior beliefs and newly acquired data, updating these beliefs to form a posterior distribution over the unknown quantities, thereby enabling adaptive and efficient experimental selection [131].

4.1 Parameters of Interest

In the context of experimental design, parameters of interest are the unknown variables or quantities that the experimenter seeks to estimate, understand, or optimize. These could be parameters of a physical system in a scientific study, model parameters in a statistical context (such as the slope parameters $\boldsymbol{\beta}$ of a linear regression [36, 84]), or any quantities that influence the behavior or predictions of a system under investigation [103, 124].

BOED aims to iteratively select data points that minimize uncertainty in these parameters of interest. In the context of the linear regression example, we aim to select the data that will best reveal what the slope parameters should be. Formally, letting ξ represent the vector of some general parameters of interest, the objective of the design process is to refine a posterior distribution $p(\xi|D)$. As each new observation is added, the posterior distribution is updated through Bayes' rule.

In BOED, a central objective is to choose data that maximizes the knowledge we expect to gain in the parameters of interest. This, in turn, allows us to make more accurate estimates of the parameters of interest. This is done by maximizing an *acquisition function*, most commonly the Expected Information Gain [93, 142, 20, 78] (IG, or occasionally EIG), over the set of candidate inputs $\mathcal{X}$. Formally, the IG is expressed as

$$IG(\boldsymbol{x}, \xi) = \mathrm{H}(p(\xi|D)) - \mathbb{E}_{y(x)}\left[\mathrm{H}(p(\xi|D \cup \{\boldsymbol{x}, y(\boldsymbol{x})\}))\right], \tag{24}$$

where $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, and where H denotes the (differential) Shannon entropy. Under the general formulation of BOED, learning procedures can be derived for general types of models and objectives, such as the noise level σ_n^2 of a GP, the weight parameters $\boldsymbol{w}$ of a Bayesian neural network [77, 71, 46], or the maximizer $\boldsymbol{x}^*$ of an unknown function [55, 56]. In Paper II, BO is formulated as a BOED objective, with the GP-induced joint distribution over the optimum $\boldsymbol{x}^*$ [56] and optimal value f^* [168] as the parameter of interest.

Paper III expands further on this idea by including relevant hyperparameters θ of the GP as additional parameters of the BOED objective to achieve a joint BO and active learning loop. Moreover, a broader class of BOED and Bayesian Active Learning (BAL) acquisition functions in the form of Statistical distance-based Active Learning (SAL), is introuced. SAL acquires data by maximizing the average disagreement between the conditional posteriors predictive distributions and the marginal posterior,

$$SAL(\boldsymbol{x}; \xi) = \mathbb{E}_\xi[d(p(y(\boldsymbol{x})|\xi, \mathcal{D}), p(y(\boldsymbol{x})|\mathcal{D}))] \tag{25}$$

which emphasizes consistent predictive performance across models rather than agreement strictly in the parameters themselves. This is similar to other prediction-oriented approaches for Bayesian active learning [12, 134], yet obtains an emphasis on the model hyperparameters. Notably, the SAL objective is equivalent to the EIG by setting d to be the forward KL divergence between the conditional posteriors and the marginal.

In Paper V, BOED is applied within the context of adaptive Group Testing (GT) to develop an algorithm tailored for feature selection in Gaussian Processes (GPs). GT is a framework for optimizing the process of identifying specific items within a set by testing groups of items rather than each one individually. In the context of Bayesian Optimal Experimental Design (BOED), GT is applied adaptively to high-dimensional feature selection, leveraging the prior and posterior distributions of a Gaussian Process (GP) to iteratively identify relevant subsets of dimensions while minimizing the number of required evaluations.

Let ξ_d denote the activeness of dimension d, where $\xi_d = 1$ indicates that the dimension is active and $\xi_d = 0$ indicates inactivity. This formulation aligns with approaches such as those in [22], which explore binary outcomes in a similar context.

By defining a *group* as a subset of dimensions ablated from their default values, and using the GP's prior signal variance σ_f^2 and noise variance σ_ε^2, the method estimates group activeness based on the signal strength within that group. The

probability of a group being active is given by:

$$p(\boldsymbol{g}_t^\intercal \boldsymbol{\xi} \geqslant 1) = \sum_{\boldsymbol{\xi} \in \{0,1\}^D} \delta_{\boldsymbol{g}_t^\intercal \boldsymbol{\xi} \geqslant 1} p(\boldsymbol{\xi}). \tag{26}$$

where $\boldsymbol{g}_t$ represents the group being evaluated, and $\delta_{\boldsymbol{g}_t^\intercal \boldsymbol{\xi} \geqslant 1}$ is the indicator function for the activeness condition. Groups are then adaptively selected to maximize the information gain over the activeness variables $\boldsymbol{\xi}$, to enable identification of relevant dimensions in high-dimensional search spaces [53].

4.2 Practical Considerations in BOED

As evidenced by Eq. (24), the strategy by which data is acquired in BOED is dependent on the parameters of interest in the problem at hand, and the model itself. Thus, the specification of the model determines the data acquisition strategy, and ultimately influences the efficiency and success of the experimental design. Moreover, the information gain in Eq. (24) is typically not computationally tractable. Practical considerations thus often entail improving computational tractability [39, 40, 41], and the dimensionality of the problem - the number of parameters of interest in the model. Calculating expected information gain or mutual information can be computationally demanding, particularly in high-dimensional settings, as it requires integration over possible outcomes for multiple realizations of the model. To manage this complexity, sampling-based techniques, such as Monte Carlo integration [140] or variational approximations [39], are frequently employed to approximate the IG. Paper V utilizes a variance-reducing Sequential Monte Carlo (SMC) approach, proposed by [22], to efficiently approximate the IG in a high-dimensional space.

5 Bayesian Optimization

Building on the general framework specified by BOED, Bayesian optimization (BO) [104, 72, 147, 144, 43, 49] is an iterative framework that utilizes a probabilistic surrogate model, typically a GP, to find the global optimum of a black-box function. As such, BO can be viewed as a specific instantiation of BOED, where the parameter of interest is the (unknown) optimum of the black-box objective. Formally, the goal is to maximize an unknown, expensive-to-evaluate D-dimensional objective function $f : \mathcal{X} \to \mathbb{R}$, $\mathcal{X} \subseteq \mathbb{R}^D$ conventionally assumed to be $[0, 1]^D$. Moreover, the objective may be noisy, so that f can only be observed through its noise-perturbed output $y(\boldsymbol{x}_i)$, where $y(\boldsymbol{x}_i) = f(\boldsymbol{x}_i) + \varepsilon_i$ and $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$. In this setting, we seek the point $\boldsymbol{x}^* \in \mathcal{X}$ such that

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}). \tag{27}$$

Since $f(\boldsymbol{x})$ is assumed to be costly to evaluate, we build a surrogate model $\hat{f}(\boldsymbol{x})$ (conventionally a GP) that provides a probabilistic estimate of $f(\boldsymbol{x})$ that can be refined over time. Using this probabilistic surrogate, BO iteratively selects points to evaluate based on an acquisition function; an expected utility criterion which quantifies candidate points in terms of either its contribution to identifying $\boldsymbol{x}^*$, or a myopic measure of quality of the candidate.

5.1 Surrogate Models for Bayesian Optimization

In Bayesian Optimization (BO), the choice of surrogate model is crucial, as it determines the ability of the optimization process to accurately approximate and explore the objective function. Surrogate models provide a probabilistic estimate of the unknown function, balancing exploration and exploitation in the search for optimal solutions. This section discusses commonly used surrogate models for BO, including Gaussian Processes (GPs) [148, 50, 7, 152, 31] and their adaptations, as well as alternative models such as Random Forests [63, 92, 114, 157, 162, 151] and Neural Networks [150, 154, 162, 151, 91, 109].

5.1.1 Gaussian Processes

There are multiple types of Gaussian processes which see frequent use in BO. Below, some of the most common types are outlined, as well as the assumptions that are associated with each type.

Vanilla Gaussian Processes Standard GPs assume a stationary kernel function and a constant-mean prior with a low level of noise. The squared exponential (RBF) [7] or Matern-5/2 kernel [149, 152] are commonly used, as they provide a degree of smoothness and is generally considered reasonable for most tasks. In some circumstances, however, model with additional assumptions with added or reduced complexity are employed, typically to enhance the performance of BO on specific types of tasks. Paper VI extensively discusses the issues related to the vanilla Gaussian process in Bayesian optimization. Specifically, it uncovers the implicit assumption of exponential complexity increase as the dimensionality of the objective is increased, and proposes a remedy in the form of a modified GP prior, defined in Eq. (19).

Additive Models Additive GPs [33, 95] assume that the function $f(x)$ can be decomposed as a sum of functions over subsets of the input dimensions:

$$f(\boldsymbol{x}) = \sum_{i=1}^{k} f_i(\boldsymbol{x}_{\mathcal{I}_i}), \tag{28}$$

where $\boldsymbol{x}_{\mathcal{I}_i}$ represents a subset of the input dimensions, and each component function is modeled as a GP. Additive models are frequently employed for high-dimensional objectives [75, 48, 181], as they reduce the overall model complexity by breaking the problem down into smaller, simpler subspaces. By isolating the influence of individual variables or groups of variables, additive models can model correlations not typically governed by the standard model, thereby improving in modeling accuracy [48] and efficiency of the optimization [75]. The fully Bayesian additive model proposed by [48] is utilized in Paper III to actively learn additive function groups in both synthetic tasks and a cosmological constant [160] learning objective.

Subspace Models Subspace models [167, 88, 115, 125, 79] aim to address the high-dimensional limitations of standard GPs by assuming that the function of interest lies in a lower-dimensional (linear) subspace of the full input space. Formally, for some $D_e \ll D$ and a projection matrix A,

$$f(\boldsymbol{x}) \approx h(\boldsymbol{z}), \quad \boldsymbol{z} = A^\top \boldsymbol{x} \tag{29}$$

where $h : \mathbb{R}^{D_e} \to \mathbb{R}$. The core idea is that only a subset of input dimensions, referred to as the active dimensions of the problem, and the number of which defines the *effective dimensionality*, significantly affect the objective. The remaining dimensions are assumed to have negligible impact. Thus, only the active dimensions require modeling. By reducing the dimensionality of the problem, subspace

models make GP-based BO more feasible in high-dimensional settings [167, 14]. A common approach to constructing subspace models is to assume the existence of an active subspace, a reduced set of dimensions that captures most of the variation in the objective function. Further refinements in subspace models include axis-aligned subspaces [37], where only specific input dimensions are assumed to be active, and random embedding approaches [167, 115, 125], where random projections are used to select relevant subspaces. The fully Bayesian, sparse axis-aligned model proposed by [37] is utilized in Paper III to systematically learn active dimensions of noisy, high-dimensional objectives.

Warped Gaussian Processes Warped GPs [149, 21] introduce flexibility by applying a transformation, or *warping*, to the input or output space, making it possible to model functions with non-stationary or non-Gaussian behavior. One common approach involves transforming the input space using a function $g(x)$ to map inputs onto a re-scaled representation that captures relevant features of the function:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(\mu(g(\boldsymbol{x})), k(g(\boldsymbol{x}), g(\boldsymbol{x}'))). \tag{30}$$

Input warping can capture complex patterns that standard GPs may miss, such as varying smoothness across the input space or highly nonlinear behavior. Warped models are particularly useful when prior knowledge suggests that certain transformations (e.g., log or exponential scales) align with the underlying behavior of the objective function. Commonly used input warpings include the Beta warping [149], or the closely related Kumaraswamy warping [21], where the normalized inputs are dimension-wise passed through the inverse CDF the aforementioned distributions. The dimension-wise Kumaraswamy warping of [21] is utilized in Paper III to better model the response surface during hyperparameter optimization of deep neural networks.

In Fig. 6, the mean predictions of a Standard (Vanilla) GP, a warped GP, and an additive GP are visualized for a 2D objective, highlighting the distinctive characteristics of each model type. A Vanilla GP generally produces conservative predictions, exhibiting smooth variation across the search space and avoiding overestimation of the magnitude of predictions. In contrast, a warped GP introduces sharper transitions and more rapid variations due to the non-stationary behavior induced by the warping functions. Finally, the dimension-wise decompositions of the additive GP enable it to extrapolate in certain unobserved regions of the search space, provided that similar values have been observed independently in each dimension.

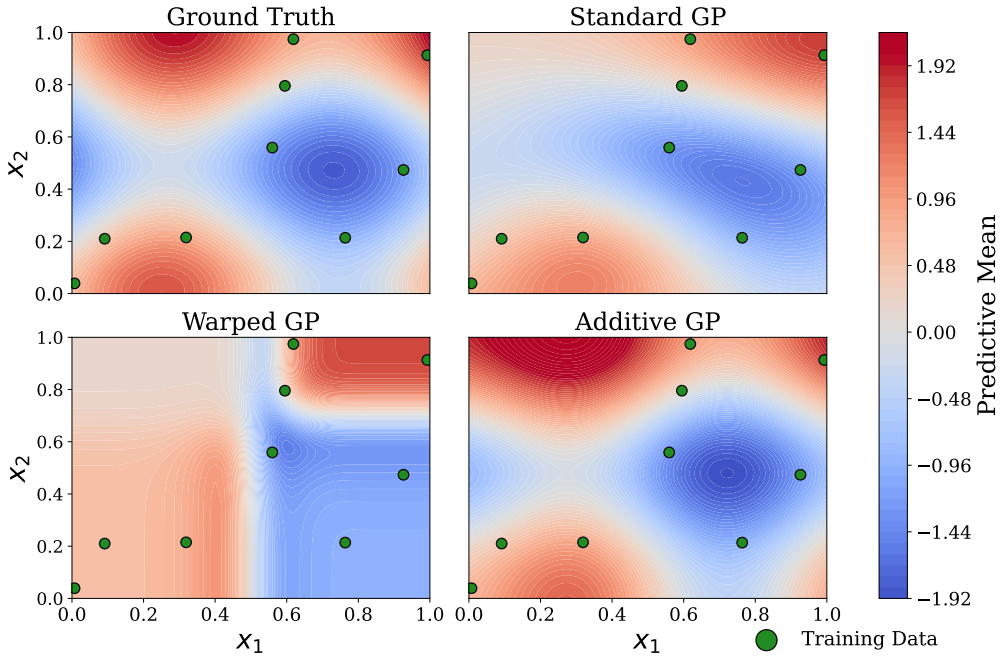

Fig. 6: Comparison of the ground truth objective and predictive means for three Gaussian Process (GP) models on a 2D synthetic function. The models include a Standard GP, a Kumaraswamy-warped GP, and an Additive GP fit with maximum likelihood estimation (MLE) and an RBF kernel. Training data points are shown in green. The color scale represents the predictive mean values. The Additive GP effectively extrapolates the region in the top left corner but slightly overestimates the magnitude of the predictions. The warped model captures steep edges and maintains relatively flat predictions in smoother regions. The standard GP, while conservative, delivers reasonably accurate predictions overall.

5.1.2 Non-Gaussian Process Models

Random Forests Random Forests (RFs)[58] offer an alternative to GPs in BO, particularly for problems involving integer-valued, categorical, or hierarchical data. An RF consists of an ensemble of decision trees, each trained on a different subset of the data, with predictions generated by averaging the outputs of the individual trees. To enable BO with RFs, modifications are introduced to quantify uncertainty, such as leveraging the variance among tree predictions as an uncertainty measure [63, 92]. This pseudo-probabilistic approach enables RFs to approximate the uncertainty estimates typically provided by GPs, while also accommodating discontinuities in the search space, which is not inherently supported by GPs. Paper I employs RFs, among other models, for BO in combination with prior-guided acquisition functions (Sec.5.2.5) to achieve efficient hyperparameter optimization under a low budget for the number of BO iterations.

Bayesian Neural Networks Bayesian Neural Networks (BNNs) [96, 117, 45] are another flexible alternative to GPs for BO, particularly in large-scale problems, where they scale more gracefully in the number of training data than GPs [150]. Unlike GPs, BNNs are parametric models, with fixed architectures that define their function space. BNNs can capture complex, non-linear relationships, but they require significant amounts of data for accurate training.

5.2 Acquisition Functions

An acquisition function is central to BO, determining the next point to evaluate. Acquisition functions are evaluated over the surrogate model to identify where sampling will most likely yield improved outcomes. Acquisition functions generally balance exploration (querying points in uncertain regions) and exploitation [49] (querying points likely to yield high objective values), and can be categorized as either myopic or non-myopic [174] based on how utility is quantified.

For a general acquisition function $\alpha(\boldsymbol{x}; \mathcal{D})$ acting on the surrogate model conditioned on the data $\mathcal{D}$ observed so far, the strategy encoded by α is obtained through its maximization over the search space [43]. Formally, we obtain the next candidate $\boldsymbol{x}_n$ to evaluate as

$$\boldsymbol{x}_n = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \alpha(\boldsymbol{x}; \mathcal{D}) \tag{31}$$

Thus, the point selected by any acquisition function is the one that yields the highest expected utility.

5.2.1 Myopic Acquisition Functions

Myopic acquisition functions [174] make decisions based on expected immediate utility of the queried candidate, without considering the impact of a sampling decision beyond the current iteration.

Expected Improvement (EI) and its extended family of acquisition functions [104, 72, 17, 47, 128, 87, 2] (EI) quantifies the nonnegative amount by which the next observation improves upon a threshold value, commonly set to the current best (either unobserved [87] or observed) value $f(\boldsymbol{x}^+)$. The standard definition of EI is expressed as

$$\alpha_{\mathrm{EI}}(\boldsymbol{x}) = \mathbb{E}\left[\max(0, f(\boldsymbol{x}) - f(\boldsymbol{x}^+))\right], \tag{32}$$

and where $f(\boldsymbol{x}^+)$ is the best observed value. For a Gaussian posterior, EI can be computed closed-form, but is frequently approximated through Monte Carlo [176, 174, 7] (MC) to allow for additional flexibility in non-standard problem settings [89, 3, 24, 4].

Upper Confidence Bound is a simple, principled acquisition function with extensive theoretical results, as it ha been shown to achieve sub-linear regret in many settings [155, 75, 11, 80]. UCB selects the next query point by considering both the predicted mean and the uncertainty. Specifically, UCB is defined as:

$$\alpha_{\mathrm{UCB}}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \kappa\sigma(\boldsymbol{x}), \tag{33}$$

where κ is a parameter that controls the balance between exploration (higher κ) and exploitation (lower κ).

Thompson Sampling [161, 137, 138](TS) is a random acquisition function which leverages the posterior distribution provided by the surrogate model to obtain draws from the probability density over the location of the optimizer $\boldsymbol{x}^*$. At each iteration, Thompson Sampling draws a sample function f_i from the posterior:

$$f(\cdot|\mathcal{D}) \sim \mathcal{GP}(\mu(\cdot), k(\cdot,\cdot)|\mathcal{D}) \tag{34}$$

and maximizes the sampled function

$$\boldsymbol{x}_n = \arg\max_{\boldsymbol{x}\in\mathcal{X}} f_i(\boldsymbol{x}; \mathcal{D}) \tag{35}$$

to obtain the next test point. As exact functions draws from the GP posterior are unattainable, TS is typically approximated either by sampling from the posterior at a finite set of input locations and taking the discrete arg max [76, 38, 101] or by optimizing continous weight-space approximation [129, 177, 8] of the function draw f_i. TS is additionally used to support multiple information-theoretic acquisition functions, outlined in Sec. 5.2.2

5.2.2 Non-myopic Acquisition Functions

Non-myopic acquisition functions take a long-term view, selecting points based on their ability to help infer the optimum after having observed their outcome. Non-myopic acquisition functions are typically more computationally demanding,

and occasionally less robust, than myopic ones. However, they can yield better optimization performance in many instances. Examples include the Entropy Search (ES) [165, 54, 118, 119] class of acquisition functions, which additionally include Predictive Entropy Search (PES) [56, 57] and Max-value Entropy Search [60, 169, 107] (MES), and the Knowledge Gradient (KG) [42, 44].

Entropy Search Acquisition Functions In the spirit of BOED, ES [54] focuses on reducing uncertainty about the location of the optimum $\boldsymbol{x}^*$ by treating it as a random variable. For each candidate point $\boldsymbol{x}$, ES evaluates the reduction in entropy over the distribution over $\boldsymbol{x}^*$ after observing $f(\boldsymbol{x})$.

$$\alpha_{\mathrm{ES}}(\boldsymbol{x}) = H(\boldsymbol{x}^*|\mathcal{D}) - \mathbb{E}_{y(\boldsymbol{x})}[H(\boldsymbol{x}^*|\mathcal{D} \cup \{(\boldsymbol{x}, y(\boldsymbol{x}))\})], \tag{36}$$

By selecting points that maximize this reduction in entropy, ES refines the search in areas that are most informative about the optimum. Other information-theoretic acquisition functions use similar criteria through re-formulation of the IG. PES [56], for example, utilizes the symmetry of the information gain [62] to express the acquisition function as a difference between entropies in the posterior predictive distribution:

$$\alpha_{\mathrm{PES}}(\boldsymbol{x}) = H(y(\boldsymbol{x})|\mathcal{D}) - \mathbb{E}_{\boldsymbol{x}^*}\left[H(y(\boldsymbol{x})|\mathcal{D}, \boldsymbol{x}^*)\right]. \tag{37}$$

MES works along similar lines as PES, but substitutes the optimal location $\boldsymbol{x}^*$ for the optimal value f^*. Entropy search acquisition functions are frequently used in multi-fidelity optimization [81, 10, 106, 107, 159] due to their fidelity-agnostic measure of utility.

Introduced in Paper II, Joint Entropy Search (JES) extends the ES family by targeting the joint information gain over both the optimal input $\boldsymbol{x}^*$ and the optimal output f^*:

$$\alpha_{JES}(\boldsymbol{x}) = I((\boldsymbol{x}, y(\boldsymbol{x})); (\boldsymbol{x}^*, f^*)|\mathcal{D}). \tag{38}$$

By conditioning on hypothetical optimal input-output pairs, JES achieves a holistic uncertainty reduction that is computationally efficient compared to ES and PES. JES relies on standard GP conditioning, sidestepping the need for costly approximations. Empirical results demonstrate that JES achieves strong performance when the underlying surrogate model is well-specified, as well as in high-noise problem settings [64].

Knowledge Gradient [42] is designed to select the next point x that provides the highest improvement in the belief about the expected maximal value of the function. Given the current data $\mathcal{D}$ and a candidate point $\boldsymbol{x}$, the KG acquisition function is defined as:

$$\alpha_{\mathrm{KG}}(\boldsymbol{x}; \mathcal{D}) = \mathbb{E}_{y(\boldsymbol{x})}\left[\max_{\boldsymbol{x}' \in \mathcal{X}} f(\boldsymbol{x}') | \mathcal{D} \cup \{(\boldsymbol{x}, y(\boldsymbol{x}))\}\right] - \max_{\boldsymbol{x}' \in \mathcal{X}} f(\boldsymbol{x}') | \mathcal{D}. \tag{39}$$

Thus, the KG value at $\boldsymbol{x}$ reflects the expected increase in the maximum posterior mean $\mathbb{E}[f]$, which may be located anywhere in the search space, if we evaluate the objective at $\boldsymbol{x}$. Similarly to EI, KG is frequently adopted in non-standard problem settings [178, 23] as a versatile look-ahead approach.

5.2.3 Monte Carlo Acquisition Functions

Monte Carlo (MC) acquisition functions [176] are a versatile and powerful class of techniques for evaluating acquisition functions. Most acquisition functions, including all the aforementioned ones, can be framed as an expectation $\mathbb{E}_{f(\boldsymbol{x})}[u(f(\boldsymbol{x})]$ or $\mathbb{E}_{y(\boldsymbol{x})}[u(y(\boldsymbol{x}))]$, over a utility function $u(f(\boldsymbol{x}))$. While some acquisition functions, such as Expected Improvement (EI) and Upper Confidence Bound (UCB), allow for closed-form solutions under Gaussian posteriors, MC acquisition functions can accomodate complex settings such as large-batch evaluations, non-Gaussian posteriors, and noisy, constrained optimization [89]. The generic formulation of a Monte Carlo acquisition function is

$$\alpha_{\mathrm{MC}}(\boldsymbol{x}; \mathcal{D}) = \mathbb{E}_{f(\boldsymbol{x})}[u(f(\boldsymbol{x}))] \approx \frac{1}{N} \sum_{i=1}^{N} u(f^{(i)}(\boldsymbol{x})), \tag{40}$$

where $\mathcal{D}$ represents the observed data, $f^{(i)}(\boldsymbol{x})$ are N samples drawn from the posterior predictive distribution of the surrogate model at $\boldsymbol{x}$, and $u(f(\boldsymbol{x}))$ is the utility function that encodes the value of evaluating $\boldsymbol{x}$.

5.2.4 Model-aware Acquisition Functions

Presented in Paper III, Self-Correcting Bayesian Optimization (SCoreBO) is the first acquisition function for BO which integrates a model hyperparameter refinement criterion into the acquisition function. By dynamically learning hyperparameters during optimization, SCoreBO addresses GP inaccuracies and adapts the model in real time, particularly in noisy or high-dimensional settings. In

practice, this is done by considering the a joint optimization- and hyperparameter learning objective building on SAL (introduced in Eq. (25)),

$$\alpha_{\mathrm{SC}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\theta}}[d(p(y(\boldsymbol{x})|\mathcal{D}, p(y(\boldsymbol{x})|\boldsymbol{\theta}, *, \mathcal{D}))], \tag{41}$$

where $*$ is some quantity related to the optimizer - either the optimal location $\boldsymbol{x}^*$, optimal value f^* or the tuple $(\boldsymbol{x}^*, f^*)$. This joint approach allows SCoreBO to achieve faster convergence and superior reliability compared to standard acquisition strategies for problems where accurate hyperparameter estimation proves difficult, such as for noisy, high-dimensional objectives.

5.2.5 Prior-weighted Acquisition Functions

Prior-weighted acquisition functions [90, 153, 1, 99] aim to incorporate additional user beliefs, such as a prior over the location over the optimal location, into the acquisition procedure. Introduced in Paper I, πBO incorporates practitioner knowledge into the acquisition process by encoding priors about promising regions in the input space in the form of probability densities $\pi(\boldsymbol{x})$. This yields a simple approach to bias a generic myopic acquisition function $\alpha(\boldsymbol{x}; \mathcal{D})$ towards a-priori promising regions of the search space,

$$\boldsymbol{x}_n \in \arg\max_{\boldsymbol{x} \in \mathcal{X}} \alpha(\boldsymbol{x}; \mathcal{D})\pi(\boldsymbol{x}). \tag{42}$$

by weighting the two terms through standard multiplication. To guard against misspecified priors, Paper I also proposes a decaying variant of the prior-weighted acquisition function

$$\alpha_{\pi,n}(\boldsymbol{x}; \mathcal{D}) \triangleq \alpha(\boldsymbol{x}, \mathcal{D})\pi_n(\boldsymbol{x}) \triangleq \alpha(\boldsymbol{x}, \mathcal{D})\pi(\boldsymbol{x})^{\beta/n} \tag{43}$$

which decreases the impact of the prior as iterations progress. In Eq. (43), β controls the rate of decay of the prior. In Paper I, it is theoretically proven that the BO strategy defined by Eq. (43) converges at standard rates when EI is employed as the base acquisition function.

Belief-weighted Monte Carlo Acquisition Functions Presented in Paper IV, belief-weighted MC acquisition functions utilize the user-specified belief over functions π to obtain sample functions $f|\pi$, which can subsequently be used to compute a π-biased MC estimate of the acquisition function:

$$\alpha_{\mathrm{MC}}(\boldsymbol{x}; \mathcal{D}) = \mathbb{E}_{(f|\pi)(\boldsymbol{x})}[u((f|\pi)(\boldsymbol{x}))] \approx \frac{1}{N}\sum_{i=1}^{N} u((f|\pi)^{(i)}(\boldsymbol{x})). \tag{44}$$

While this approach yields less refined MC estimates compared to state-of-the-art MC acquisition function techniques [7], its ability to incorporate arbitrary priors over functions offers significant practical advantages in terms of user customization. By aligning the acquisition process with practitioner-defined beliefs, belief-weighted MC acquisition functions enable greater flexibility and adaptability to real-world optimization scenarios. Previous works, that have incorporated various forms of user beliefs by adjusting the surrogate model can be found in [120, 69].

5.3 The BO Loop: An Iterative Trial-and-Error Procedure

BO is inherently an iterative process, designed to refine its understanding of an objective function through a cycle of informed experimentation. At each iteration, the surrogate model $\hat{f}(\boldsymbol{x})$ is updated and re-fit based on the most recent observations. This model adjustment provides an updated probabilistic approximation of the objective function, giving a refined prediction of the mean and variance across the search space, as well as a refined belief over the hyperparameters of the model itself. Subsequently, the acquisition function can be computed using the surrogate model's updated predictions, so that the search strategy with the most current information available.

With the model updated, the BO algorithm proceeds to identify the point that maximizes the acquisition function across the search space. This selection process, which determines where the next evaluation should occur. Once the point is selected, it is evaluated on the true objective function. This evaluation, though costly, serves as a new piece of knowledge in the BO loop. By observing the outcome at this chosen point, BO effectively performs a controlled "trial" within the trial-and-error framework. The new data point, comprising the selected input and its observed output, is then added to the dataset. With this updated dataset, BO now has an enriched set of information that reflects both past and newly acquired knowledge of the function.

The iterative process continues, each loop enhancing the model's ability to make informed decisions. BO stops iterating once a pre-defined criterion is satisfied. This stopping criterion could be based on reaching a maximum number of evaluations, detecting convergence in the observed objective values, or determining that additional evaluations are unlikely to yield significant improvements. When the process halts, BO returns the best solution found during the iterations, offering an optimized result while maintaining a minimal number of evaluations.

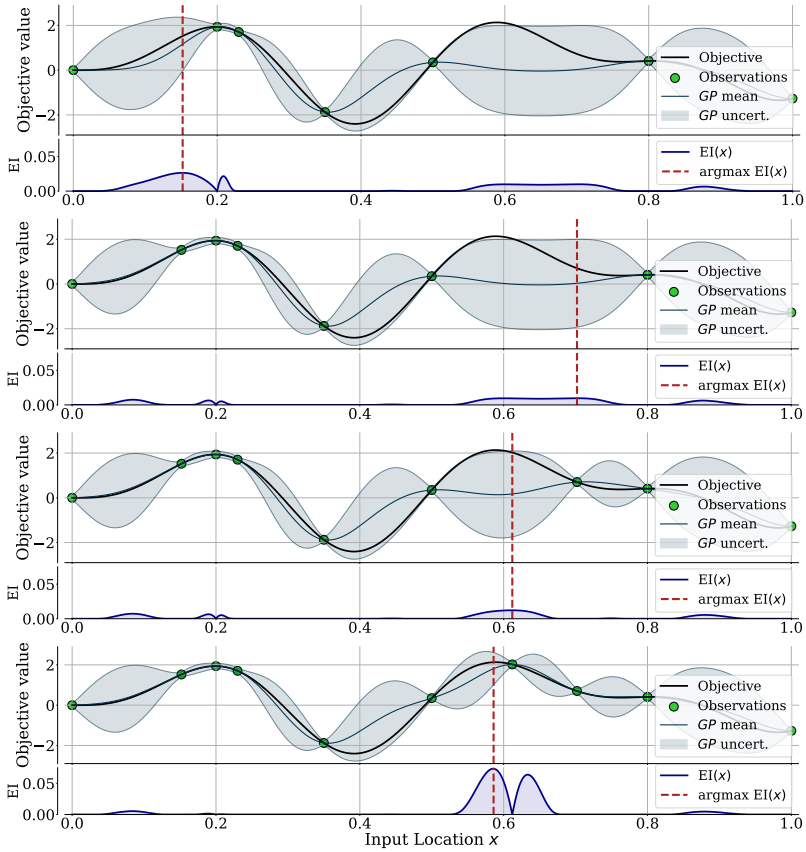

Fig. 7: Four iterations of BO using EI as the acquisition function, and a Gaussian Process with an RBF kernel as the surrogate model. (Top) The GP surrogate model in grey, the observed data in green, and the true, unobserved objective function in black. (bottom) The EI acquisition function in navy blue, and its argmax, corresponding to the upcoming query, in red. The optimization loop interleaves exploration and exploitation as iterations progress, eventually testing points close to the global maximizer of the objective.

Algorithm 1 Bayesian Optimization Loop

Require: Objective function $f(\boldsymbol{x})$, search space $\mathcal{X}$, acquisition function $\alpha(\boldsymbol{x}; \mathcal{D})$, initial data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$

while stopping criterion not met **do**
  Fit surrogate model $\hat{f}$ on data $\mathcal{D}$
  Define acquisition function $a(\boldsymbol{x}; \mathcal{D})$ based on surrogate model $\hat{f}$
  Select next point to evaluate: $\boldsymbol{x}_{t+1} = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \alpha(\boldsymbol{x}; \mathcal{D})$
  Evaluate objective function: $y_{t+1} = f(\boldsymbol{x}_{t+1}) + \varepsilon_{t+1}$
  Update dataset: $\mathcal{D} = \mathcal{D} \cup \{(\boldsymbol{x}_{t+1}, y_{t+1})\}$
end while

return Best point $\boldsymbol{x}_n^* = \arg\max_{\boldsymbol{x} \in \mathcal{D}} y(\boldsymbol{x})$

5.3.1 The Interplay Between Model and Acquisition

A critical feature of BO is the intricate interplay between the surrogate model and the acquisition function [98, 52]. This relationship forms a feedback loop in which each component dynamically informs the other. The surrogate model provides the probabilistic representation of the objective function, influencing the acquisition function's ability to prioritize candidate points. Conversely, the acquisition function guides the selection of new data points, which directly shape the subsequent updates to the model.

This mutual dependency is at the heart of BO's iterative nature, and is one where pathologies and inefficiencies may occur [121, 158]. The model's predictive accuracy governs the utility of the acquisition function, as poorly fit models may misrepresent uncertainty by means of ill-chosen hyperparameters. This interplay becomes particularly evident in challenging scenarios such as high-dimensional optimization or when dealing with noisy objectives. For example, in Paper VI, the proposed framework demonstrates how simplifying the model's complexity - by adjusting the kernel's lengthscale prior - improves BO performance across dimensionalities and stabilizes the GP fit. Here, the model's reduced assumptions facilitate an appropriately simple representation of the objective. Thus, the model is less inclined to over-explore the boundary due to exceedingly long lengthscales, or overly exploit due to a lack of correlation in the data.

Conversely, Paper III highlights how acquisition functions can explicitly contribute to improving the model itself, thus acting synergistically in the interplay between the two components. The SCoreBO framework integrates model hyperparameter learning into the acquisition process, dynamically refining the surrogate model's predictions during the optimization loop. Paper II proposes a more lightweight γ-exploit approach to guard against model misspecification in the acquisition function, where with probability $\gamma \in [0, 1]$, the maximizer of the posterior mean, $\arg\max_{\boldsymbol{x} \in \mathcal{X}} \mu(\boldsymbol{x})$ is chosen as the next query instead of the regular maximizer of the acquisition function.

The interplay between model and acquisition function is, at large, a fairly unexplored area of research. With few exceptions [180, 171], works on BO's theoretical convergence guarantees [155, 17, 27, 79, 80, 15, 11] generally presume that the hyperparameters of the model are fixed, thus removing the complex practical interplay between the two components from the analysis. Moreover, non-standard models tailored towards specific problem settings [37, 21, 48], such as those outlined in Sec. 5.1.1 implicitly assume that the data collected through the BO procedure will suffice in order to obtain a model of acceptable accuracy. While this assumption demonstrably holds on many relevant problems [37, 48],

Paper III demonstrates that under particular circumstances, such as significant observation noise, this assumption does not hold.

6 Contributions

This thesis contributes to advancing Bayesian Optimization by addressing three core challenges related to user knowledge and information: user-guided optimization, auxiliary model-level objectives, and simplifying assumptions for efficient high-dimensional optimization. Each paper targets a specific aspect of these challenges, offering novel methods and practical tools that extend the utility of BO across varied and complex problem domains. Together, the papers constitute a substantial effort towards improving BO's efficiency and adaptability to both user-defined and algorithmic assumptions.

6.1 RQ1: User-Guided Bayesian Optimization

Two complementary approaches to this problem are presented in Paper I and Paper IV, each offering distinct methods for embedding user knowledge into the BO framework. Together, they demonstrate the benefits of leveraging practitioner insights in diverse and flexible ways while maintaining theoretical rigor and practical utility.

Paper I introduces πBO, a simple and intuitive method that allows practitioners to define prior beliefs about promising regions of the input space in the form of probability distributions. By incorporating these priors into the acquisition function, πBO enables the optimizer to focus its search on regions that are more likely to contain the optimum, based on the practitioner's prior knowledge. This approach is particularly effective in scenarios like hyperparameter tuning for machine learning models, where practitioners often have prior intuition about optimal ranges for parameters such as learning rates or regularization strengths. The method is computationally lightweight, seamlessly integrates into existing BO frameworks, and retains theoretical regret bounds when used with the Expected Improvement (EI) [104] acquisition function. Empirical results demonstrate that πBO significantly accelerates convergence in real-world tuning tasks, reducing the number of evaluations needed to achieve high-performance solutions.

While πBO focuses on spatial priors over the input space and their integration into the *acquisition function*, Paper IV broadens the scope of user-guided BO by introducing Collaborative Bayesian Optimization (ColaBO), a framework designed to accommodate a wider variety of practitioner insights through integration with the *surrogate model*. Beyond spatial priors, ColaBO allows users to encode more complex forms of prior knowledge, such as bounds on the optimal

value, relational preferences between candidates, and specific constraints on achievable outcomes. ColaBO achieves this by reshaping the GP surrogate model itself to reflect these user-defined priors through sample-wise Bayesian posterior updating in accordance with the aforementioned prior. This enables a more comprehensive alignment between the optimization process and practitioner expectations. Unlike πBO, which is exclusively suited for guiding the optimizer within the input space, ColaBO provides a more flexible framework capable of handling diverse prior information. This generality comes with increased algorithmic complexity, but it allows ColaBO to address a broader range of real-world scenarios where user insights extend beyond spatial expectations.

6.2 RQ2: Leveraging Auxiliary Model-Level Objectives

The efficiency and reliability of BO heavily depend on the accuracy of the surrogate model, which approximates the underlying objective function. Auxiliary information or parameters, such as the maximal value or the model hyperparameters, offer a critical pathway to enhance this accuracy during the optimization process. However, traditional BO frameworks may disregard the active learning of these parameters along with the optimizer, disregarding the potential benefit that learning these quantities will have on the learning of the location of the optimizer. Paper II and Paper III address this challenge by proposing methods to dynamically validate and exploit auxiliary information, improving both the model's predictive performance and the overall optimization process.

Paper II introduces JES, an information-theoretic acquisition function that reduces uncertainty in both optimal input location and output value simultaneously. Unlike traditional methods such as ES [54] or PES [56], which focus solely on the uncertainty of the optimum's location, JES models the joint probability distribution of both the optimal input and output values. By conditioning on hypothetical optimal input-output pairs, JES captures a holistic view of uncertainty related to the optimizer. This joint modeling allows JES to bypass the complex and computationally intensive approximations that are prominent in both ES and PES, instead relying on standard GP conditioning techniques. The result is a method that achieves state-of-the-art performance while remaining computationally efficient, offering improvements in both sample efficiency and practical usability across diverse optimization tasks. JES demonstrates how auxiliary information about the output space, when combined with the input space, can unlock additional efficiency in BO.

Expanding on this foundation, Paper III focuses on the reliability of the surrogate model by addressing the uncertainty in GP hyperparameters. Hyperparameters

such as lengthscales, signal variance, and noise level are generally estimated through Maximum Likelihood, Maximum a Posteriori or fully Bayesian methods, and are presumed to be estimated accurately. However, inaccuracies in these estimates can lead to poor model predictions and suboptimal optimization performance, as the objective function will not be faithfully modeled. To address this, the paper introduces SAL, a method for refining GP hyperparameters dynamically during the optimization process. SAL measures statistical distances between candidate predictive posteriors to prioritize data acquisition that improves hyperparameter learning, contributing to better model accuracy and reliability.

Building on SAL, the paper proposes SCoreBO, a novel acquisition function that integrates JES, and other information-theoretic BO approaches, with active hyperparameter learning in the spirit of BOED. SCoreBO dynamically balances the dual objectives of locating the optimizer and improving the surrogate model, allowing the BO framework to adaptively refine its predictions over time. This approach is particularly effective in challenging applications, such as those involving noisy high-dimensional tasks, non-stationary objectives or additively decomposable functions. By focusing on both the primary optimization objective and the auxiliary task of hyperparameter refinement, SCoreBO achieves faster convergence and higher reliability across a range of standard and complex tasks. It enhances the performance of non-standard GP models, such as SAASBO [37] and HEBO [21], and identifies relevant substructures in additively decomposable objectives.

6.3 RQ3: High-Dimensional Bayesian Optimization

Paper V and Paper VI approach high-dimensional BO from complementary perspectives, offering distinct strategies for managing high-dimensional problems. Together, they highlight the critical trade-offs between employing explicit structural assumptions and reducing model complexity implicitly, without making structural trade-offs in the form of additive decomposability or effective subspaces.

Paper V emphasizes the approach of structural assumptions in the form of effective subspaces: by assuming that only a subset of the input dimensions significantly influences the optimization objective. This assumption is common in high-dimensional BO but is rarely exploited to its full extent. The paper leverages Group Testing (GT) theory to efficiently identify the active dimensions in the input space. Originally developed to isolate key elements within a larger set through grouped queries, GT is adapted here for noisy and continuous

settings. The resulting Group Testing Bayesian Optimization (GTBO) algorithm operates in two stages: a feature selection phase to identify the active subspace, followed by a low-dimensional BO phase focused exclusively on the relevant dimensions. This methodology, based on extreme structural assumptions, ensures sample efficiency, given that these assumptions hold. Moreover, it preserves interpretability, allowing practitioners to gain insights into which dimensions are most impactful. Theoretical contributions include extending GT to continuous feature selection, while empirical results demonstrate that GTBO performs competitively with state-of-the-art high-dimensional methods across diverse benchmarks.

Paper VI, on the other hand, challenges the reliance on structural assumptions by proposing a fundamentally different approach. Rather than focusing on subspaces or feature selection, this work examines the role of model complexity in high-dimensional BO. The paper hypothesizes that the difficulties faced by BO in high-dimensional settings arise not from the dimensionality itself but from overly complex assumptions about the function's behavior. By adjusting the GP kernel's length-scale prior to scale with dimensionality, the algorithm simplifies the surrogate model while retaining its global applicability. This approach, referred to as the Dimension-Scaled Prior (DSP) or *Vanilla BO* due to its similarity to the standard algorithm, eliminates the need for predefined structural constraints or active subspace assumptions. The lightweight modification transforms standard BO into a competitive method for high-dimensional tasks, offering a plug-and-play solution that performs effectively on a range of real-world problems. Results show that this simplified approach often outperforms more specialized high-dimensional methods, particularly in settings where structural assumptions cannot be verified.

The two methods address distinct but complementary aspects of high-dimensional BO. Paper VI demonstrates the power of structured assumptions when relevant subspaces can be identified, offering high sample efficiency and actionable insights about the problem's structure. By contrast, Paper VI provides a general-purpose solution that avoids relying on unverified assumptions, making it robust across a broader range of problem settings. Together, these contributions underscore the importance of low-complexity modeling assumptions, specifically related to the complexity of the objective, in high-dimensional optimization and in BO generally. Paper V tests the boundaries of structural assumptions on the surrogate for high-dimensional optimization. Paper VI marks a significant step forward in making Bayesian Optimization more reliable for high-dimensional problems, enabling it to tackle increasingly complex and resource-intensive optimization tasks without resorting to restrictive or complex assumptions. Moreover, the

DSP has been adopted as the default hyperparameter prior by the BoTorch [7] BO research framework, as well as the production-grade BO tools Ax [6] and BoFire [31].

7 Conclusions, Outlook and Future Work

As the prevalence of BO increases in applications across engineering, scientific discovery, and machine learning, we require methods that are not only efficient but also adaptable, robust and user-centric. This thesis addresses key challenges in BO by advancing its adaptability to real-world demands through user-guided insights, refined surrogate modeling, and innovative approaches to high-dimensional problems. By integrating practitioner expertise - such as spatial priors and relational preferences - BO is transformed into a collaborative, intuitive tool that aligns closely with domain-specific needs. The research also enhances the reliability of surrogate models by dynamically refining assumptions about complex, noisy, and high-dimensional objectives, ensuring greater efficiency and robustness. Finally, it redefines high-dimensional optimization by balancing structured methods like subspace identification with assumption-light approaches, demonstrating how thoughtful modeling choices can enable BO to tackle increasingly complex tasks with efficiency and precision.

Looking ahead, several promising directions for future research emerge:

User Knowledge Integration While both Paper I and Paper IV tackle the integration of auxiliary knowledge in novel ways, they both share limitations distinct to user-guided approaches. Firstly, the elicitation of suitable user knowledge is seldom trivial. Whether providing an input distribution, an optimal value or a collection of preference relations, practitioners are may face challenges when asked to quantify their beliefs. As such, tools that assist in knowledge elicitation [70] are pivotal towards BO with maximal knowledge integration. Secondly, the biasing of the search space that the injection of user knowledge entails can substantially accelerate optimization if accurate, but will inherently risk a worsening of finite-time performance if inaccurate. Future work should address these challenges, as well as the integration of prior beliefs through amortized approaches [108, 109] or by imposing structural model-level assumptions [120, 69], that balance the simplicity of Paper I with the generality of Paper IV.

Balancing Auxiliary Objectives Paper II and Paper III, demonstrate that, by going beyond the primary objective of locating the optimizer, efficiency gains can be achieved through the acquisition of supplementary information. The natural drawback of these approaches is apparent: when auxiliary information is not helpful towards locating the optimizer, sample efficiency may deteriorate. Moreover, there may be instances where, despite being helpful, the task of obtaining relevant auxiliary information may be as challenging as the task of locating the optimizer. Future research should explore how to appropriately

determine how to properly balance the acquisition of auxiliary information against the main task of locating the optimizer. Additionally, hyperparameter uncertainty may serve as a higher-order criterion for early stopping [97, 175] and for initialization strategies, to better encompass various types of uncertainty through the entirety of the BO loop.

High-Complexity Optimization Paper VI demonstrates that sensible complexity assumptions are pivotal for BO performance - even more so than any given structural assumption. Future work should examine when various structural assumptions, such as additivity or an active subspace, are appropriate, and how to adapt the methodology from Paper VI to non-conventional search spaces. Potential areas for exploration include, but are not limited to, discrete and combinatorial optimization problems [28, 122, 29, 25, 30, 126], as well as domains with non-Euclidean and structured search spaces [51, 105, 156].

In summary, this thesis represents a significant step toward a robust and user-aligned vision for BO. By enhancing its ability to incorporate user knowledge, actively obtain and exploit auxilliary information, and adapt to complexity, this work brings BO closer to becoming a universally reliable and efficient framework for solving the most demanding practical optimization challenges - both independently and in collaboration with human expertise.

References

[1] Masaki Adachi, Satoshi Hayakawa, Saad Hamid, Martin Jørgensen, Harald Oberhauser, and Micheal A. Osborne. Sober: Highly parallel bayesian optimization and bayesian quadrature over discrete and mixed spaces, 2023. URL https://arxiv.org/abs/2301.11832.

[2] Sebastian Ament, Sam Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for bayesian optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1vyAG6j9PE.

[3] Raul Astudillo and Peter Frazier. Bayesian optimization of composite functions. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learningbl Research*, pages 354–363. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/astudillo19a.html.

[4] Raul Astudillo and Peter Frazier. Bayesian optimization of function networks. *Advances in neural information processing systems*, 34:14463–14475, 2021.

[5] F. Bach and D. Blei, editors. *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, volume 37, 2015. Omnipress.

[6] Eytan Bakshy, Lili Dworkin, Brian Karrer, Konstantin Kashin, Benjamin Letham, Ashwin Murthy, and Shaun Singh. AE: A domain-agnostic platform for adaptive experimentation. In *Conference on Neural Information Processing Systems*, pages 1–8, 2018.

[7] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. In *Advances in Neural Information Processing Systems*, 2020. URL http://arxiv.org/abs/1910.06403.

[8] Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 462–471. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/baptista18a.html.

[9] P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors. *Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems (NeurIPS'12)*, 2012.

[10] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Multi-fidelity multi-objective bayesian optimization: An output space entropy search approach. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 34, pages 10035–10043, 2020.

[11] Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. No-regret bayesian optimization with unknown hyperparameters. *Journal of Machine Learning Research*, 20(50):1–24, 2019. URL `http://jmlr.org/papers/v20/18-213.html`.

[12] Freddie Bickford Smith, Andreas Kirsch, Sebastian Farquhar, Yarin Gal, Adam Foster, and Tom Rainforth. Prediction-oriented bayesian active learning. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 7331–7348. PMLR, 25–27 Apr 2023. URL `https://proceedings.mlr.press/v206/bickfordsmith23a.html`.

[13] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 2018.

[14] Mickaël Binois and Nathan Wycoff. A survey on high-dimensional gaussian process modeling with application to bayesian optimization. *ACM Trans. Evol. Learn. Optim.*, 2(2), aug 2022. doi: 10.1145/3545611. URL `https://doi.org/10.1145/3545611`.

[15] Ilija Bogunovic and Andreas Krause. Misspecified gaussian process bandit optimization. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=kbzx0uNZdS`.

[16] X. Bouthillier and G. Varoquaux. Survey of machine-learning experimental methods at NeurIPS2019 and ICLR2020. Research report, Inria Saclay Ile de France, 2020.

[17] Adam D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12:2879–2904, 2011.

[18] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. Deisenroth. Bayesian gait optimization for bipedal locomotion. In P. Pardalos and M. Resende, editors, *Proceedings of the Eighth International Conference on Learning and Intelligent Optimization (LION'14)*, Lecture Notes in Computer Science. Springer, 2014.

[19] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995. ISSN 08834237. URL `http://www.jstor.org/stable/2246015`.

[20] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

[21] Alexander Imani Cowen-Rivers, Wenlong Lyu, Zhi Wang, Rasul Tutunov, Jianye Hao, Jun Wang, and Haitham Bou-Ammar. HEBO: heteroscedastic evolutionary bayesian optimisation. *CoRR*, abs/2012.03826, 2020. URL `https://arxiv.org/abs/2012.03826`.

[22] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Noisy Adaptive Group Testing using Bayesian Sequential Experimental Design. *CoRR*, abs/2004.12508, 2020.

[23] Sam Daulton, Maximilian Balandat, and Eytan Bakshy. Hypervolume knowledge gradient: A lookahead approach for multi-objective Bayesian optimization with partial information. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 7167–7204. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/daulton23a.html`.

[24] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9851–9864. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/6fec24eac8f18ed793f5eaad3dd7977c-Paper.pdf`.

[25] Samuel Daulton, Xingchen Wan, David Eriksson, Maximilian Balandat, Michael A Osborne, and Eytan Bakshy. Bayesian optimization over discrete and mixed spaces via probabilistic reparameterization. *Advances in Neural Information Processing Systems*, 35:12760–12774, 2022.

[26] Alp Kucukelbir David M. Blei and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773. URL https://doi.org/10.1080/01621459.2017.1285773.

[27] Nando de Freitas, Alex Smola, and Masrour Zoghi. Exponential regret bounds for Gaussian process bandits with deterministic observations. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML'12)*. Omnipress, 2012.

[28] Aryan Deshwal and Jana Doppa. Combining latent space and structured kernels for bayesian optimization over combinatorial spaces. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8185–8200. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/44e76e99b5e194377e955b13fb12f630-Paper.pdf.

[29] Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Bayesian optimization over hybrid spaces. In *International Conference on Machine Learning*, pages 2632–2643. PMLR, 2021.

[30] Aryan Deshwal, Sebastian Ament, Maximilian Balandat, Eytan Bakshy, Janardhan Rao Doppa, and David Eriksson. Bayesian optimization over high-dimensional combinatorial spaces via dictionary-based embeddings. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 7021–7039. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/deshwal23a.html.

[31] Johannes P. Dürholt, Thomas S. Asche, Johanna Kleinekorte, Gabriel Mancino-Ball, Benjamin Schiller, Simon Sung, Julian Keupp, Aaron Osburg, Toby Boyne, Ruth Misener, Rosona Eldred, Wagner Steuer Costa, Chrysoula Kappatou, Robert M. Lee, Dominik Linzner, David Walz, Niklas Wulkow, and Behrang Shafei. Bofire: Bayesian optimization framework intended for real experiments, 2024. URL https://arxiv.org/abs/2408.05040.

[32] David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, Apollo - University of Cambridge Repository, 2014. URL https://www.repository.cam.ac.uk/handle/1810/247281.

[33] David K Duvenaud, Hannes Nickisch, and Carl Rasmussen. Additive gaussian processes. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/4c5bde74a8f110656874902f07378009-Paper.pdf.

[34] Afonso Eduardo and Michael U. Gutmann. Bayesian optimization with informative covariance. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=JwgVBv18RG.

[35] Adel Ejjeh, Leon Medvinsky, Aaron Councilman, Hemang Nehra, Suraj Sharma, Vikram Adve, Luigi Nardi, Eriko Nurvitadhi, and Rob A Rutenbar. HPVM2FPGA: Enabling True Hardware-Agnostic FPGA Programming. In *Proceedings of the 33rd IEEE International Conference on Application-specific Systems, Architectures, and Processors*, 2022.

[36] G. Elfving. Optimum allocation in linear regression theory. *The Annals of Mathematical Statistics*, 23(2):255–262, 1952. ISSN 00034851. URL http://www.jstor.org/stable/2236451.

[37] David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pages 493–503. PMLR, 2021.

[38] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/6c990b7aca7bc7058f5e98ea909e924b-Paper.pdf.

[39] Adam Foster, Martin Jankowiak, Elias Bingham, Paul Horsfall, Yee Whye Teh, Thomas Rainforth, and Noah Goodman. Variational bayesian optimal experimental design. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/d55cbf210f175f4a37916eafe6c04f0d-Paper.pdf.

[40] Adam Foster, Desi R Ivanova, Ilyas Malik, and Tom Rainforth. Deep adaptive design: Amortizing sequential bayesian experimental design. In

Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3384–3395. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/foster21a.html.

[41] AE Foster. *Variational, Monte Carlo and policy-based approaches to Bayesian experimental design*. PhD thesis, University of Oxford, 2021.

[42] P. Frazier, W. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM J. Control and Optimization*, 47: 2410–2439, 01 2008. doi: 10.1137/070693424.

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

[44] Peter I Frazier and Jialei Wang. Bayesian optimization for materials design. In *Information science for materials discovery and design*, pages 45–75. Springer, 2016.

[45] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

[46] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/gal17a.html.

[47] J. Gardner, M. Kusner, Z. Xu, K. Weinberger, and J. Cunningham. Bayesian Optimization with Inequality Constraints. In Xing and Jebara [179], pages 937–945.

[48] J. Gardner, C. Guo, K. Weinberger, R. Garnett, and R. Grosse. Discovering and Exploiting Additive Structure for Bayesian Optimization. In A. Singh and J. Zhu, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 1311–1319. Proceedings of Machine Learning Research, 2017.

[49] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023. to appear.

[50] The GPyOpt-authors. GPyOpt: A bayesian optimization framework in python. http://github.com/SheffieldML/GPyOpt, 2016.

[51] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 2020.

[52] Huong Ha, Vu Nguyen, Hung Tran-The, Hongyu Zhang, Xiuzhen Zhang, and Anton van den Hengel. Provably efficient bayesian optimization with unknown gaussian process hyperparameter estimation, 2024. URL https://arxiv.org/abs/2306.06844.

[53] Erik Orm Hellsten, Carl Hvarfner, Leonard Papenmeier, and Luigi Nardi. High-dimensional bayesian optimization with group testing, 2023.

[54] P. Hennig and C. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 98888(1):1809–1837, 2012.

[55] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(1):1809–1837, June 2012. ISSN 1532-4435.

[56] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, 2014. URL https://proceedings.neurips.cc/paper/2014/file/069d3bb002acd8d7dd095917f9efe4cb-Paper.pdf.

[57] José Miguel Hernández-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In *International conference on machine learning*, pages 1699–1707. PMLR, 2015.

[58] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, 1995. doi: 10.1109/ICDAR.1995.598994.

[59] Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014. URL http://jmlr.org/papers/v15/hoffman14a.html.

[60] Matthew W. Hoffman and Zoubin Ghahramani. Output-space predictive entropy search for flexible global optimization. In *NeurIPS workshop on Bayesian optimization*, 2016.

[61] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The Annals of Statistics*, 2008.

[62] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[63] F. Hutter, H. Hoos, K. Leyton-Brown, and K. Murphy. Time-bounded sequential parameter optimization. In C. Blum, editor, *Proceedings of the Fourth International Conference on Learning and Intelligent Optimization (LION'10)*, volume 6073 of *Lecture Notes in Computer Science*, pages 281–298. Springer, 2010.

[64] Carl Hvarfner, Frank Hutter, and Luigi Nardi. Joint entropy eearch for maximally-informed bayesian optimization. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022.

[65] Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. PiBO: Augmenting Acquisition Functions with User Beliefs for Bayesian Optimization. In *International Conference on Learning Representations*, 2022.

[66] Carl Hvarfner, Erik Hellsten, Frank Hutter, and Luigi Nardi. Self-correcting bayesian optimization through bayesian active learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=dX9MjUtP1A`.

[67] Carl Hvarfner, Erik O. Hellsten, and Luigi Nardi. Vanilla bayesian optimization performs great in high dimensions. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[68] Carl Hvarfner, Frank Hutter, and Luigi Nardi. A general framework for user-guided bayesian optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=NjU0jtXcYn`.

[69] Taewon Jeong and Heeyoung Kim. Objective bound conditional gaussian process for bayesian optimization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4819–4828.

PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/jeong21a.html.

[70] Zhiyuan Jerry Lin, Raul Astudillo, Peter Frazier, and Eytan Bakshy. Preference exploration for efficient bayesian optimization with multiple outcomes. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 4235–4258. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/jerry-lin22a.html.

[71] Andrew Jesson, Panagiotis Tigas, Joost van Amersfoort, Andreas Kirsch, Uri Shalit, and Yarin Gal. Causal-bald: Deep bayesian active learning of outcomes to infer treatment-effects from observational data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 30465–30478. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/ffa4eb0e32349ae57f7a0ee8c7cd7c11-Paper.pdf.

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

[73] Donald R Jones. Large-scale multi-disciplinary mass optimization in the auto industry. In *MOPTA 2008 Conference (20 August 2008)*, 2008.

[74] A G Journel and C J Huijbregts. Mining geostatistics, Jan 1976.

[75] K. Kandasamy, J. Schneider, and B. Póczos. High Dimensional Bayesian Optimisation and Bandits via Additive Models. In Bach and Blei [5], pages 295–304.

[76] K. Kandasamy, A. Krishnamurthy, J. Schneider, and B. Póczos. Parallelised Bayesian optimisation via Thompson sampling. In A. Storkey and F Perez-Cruz, editors, *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84, pages 133–142. Proceedings of Machine Learning Research, 2018.

[77] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.,

2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/95323660ed2124450caaac2c46b5ed90-Paper.pdf.

[78] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019.

[79] Johannes Kirschner, Mojmir Mutny, Nicole Hiller, Rasmus Ischebeck, and Andreas Krause. Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3429–3438. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kirschner19a.html.

[80] Johannes Kirschner, Ilija Bogunovic, Stefanie Jegelka, and Andreas Krause. Distributionally robust bayesian optimization. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2174–2184. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/kirschner20a.html.

[81] Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial intelligence and statistics*, pages 528–536. PMLR, 2017.

[82] Mario Köppen. The curse of dimensionality. *5th online world conference on soft computing in industrial applications (WSC5)*, 2000.

[83] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In Bartlett et al. [9], pages 1097–1105.

[84] Warren F. Kuhfeld, Randall D. Tobias, and Mark J. Garratt. Efficient experimental design with marketing research applications. *Journal of Marketing Research*, 31:545 – 557, 1994. URL https://api.semanticscholar.org/CorpusID:12089972.

[85] Vidhi Lalchand and Carl Edward Rasmussen. Approximate inference for fully bayesian gaussian process regression. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–12. PMLR, 2020.

[86] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[87] B. Letham, K. Brian, G. Ottoni, and E. Bakshy. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 2018.

[88] Ben Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional Bayesian optimization. *Advances in neural information processing systems*, 33:1546–1558, 2020.

[89] Benjamin Letham, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy. Constrained bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 2019.

[90] C. Li, S. Rana, S. Gupta, V. Nguyen, S. Venkatesh, A. Sutti, D. R. de Celis, T. Slezak, M. Height, M. Mohammed, and I. Gibson. Accelerating experimental design by incorporating experimenter hunches. In *IEEE International Conference on Data Mining, ICDM*, pages 257–266. IEEE Computer Society, 2018.

[91] Yucen Lily Li, Tim G. J. Rudner, and Andrew Gordon Wilson. A study of bayesian neural network surrogates for bayesian optimization, 2024.

[92] Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. Smac3: A versatile bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23 (54):1–9, 2022. URL `http://jmlr.org/papers/v23/21-0888.html`.

[93] D. V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956. ISSN 00034851, 21688990. URL `http://www.jstor.org/stable/2237191`.

[94] Sanae Lotfi, Pavel Izmailov, Gregory Benton, Micah Goldblum, and Andrew Gordon Wilson. Bayesian model selection, the marginal likelihood, and generalization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 14223–14247. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/lotfi22a.html`.

[95] Xiaoyu Lu, Alexis Boukouvalas, and James Hensman. Additive Gaussian processes revisited. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 14358–14383. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/lu22b.html`.

[96] David J.C MacKay. Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 354(1): 73–80, 1995. ISSN 0168-9002. doi: https://doi.org/10.1016/0168-9002(94)00931-7. URL https://www.sciencedirect.com/science/article/pii/0168900294009317. Proceedings of the Third Workshop on Neutron Scattering Data Analysis.

[97] Anastasia Makarova, Huibin Shen, Valerio Perrone, Aaron Klein, Jean Baptiste Faddoul, Andreas Krause, Matthias Seeger, and Cedric Archambeau. Automatic termination for hyperparameter optimization. In Isabelle Guyon, Marius Lindauer, Mihaela van der Schaar, Frank Hutter, and Roman Garnett, editors, *Proceedings of the First International Conference on Automated Machine Learning*, volume 188 of *Proceedings of Machine Learning Research*, pages 7/1–21. PMLR, 25–27 Jul 2022. URL https://proceedings.mlr.press/v188/makarova22a.html.

[98] Gustavo Malkomes, Chip Schaff, and Roman Garnett. Bayesian optimization for automated model selection. In Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors, *Proceedings of the Workshop on Automatic Machine Learning*, volume 64 of *Proceedings of Machine Learning Research*, pages 41–47, New York, New York, USA, 24 Jun 2016. PMLR. URL https://proceedings.mlr.press/v64/malkomes_bayesian_2016.html.

[99] Neeratyoy Mallik, Edward Bergman, Carl Hvarfner, Danny Stoll, Maciej Janowski, Marius Lindauer, Luigi Nardi, and Frank Hutter. Priorband: Practical hyperparameter optimization in the age of deep learning. *arXiv preprint 2306.12370*, 2023.

[100] B. Matérn. Spatial variation. *Meddelanden fran Statens Skogsforskningsinstitut*, 1960.

[101] Natalie Maus, Haydn Jones, Juston Moore, Matt J Kusner, John Bradshaw, and Jacob Gardner. Local latent space bayesian optimization over structured inputs. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 34505–34518. Curran Associates, Inc., 2022.

[102] Matthias Mayr, Carl Hvarfner, Konstantinos Chatzilygeroudis, Luigi Nardi, and Volker Krueger. Learning skill-based industrial robot tasks with user priors. *IEEE 18th International Conference on Automation Science and Engineering*, 2022. URL https://arxiv.org/abs/2208.01605.

[103] J. A. Melendez, R. J. Furnstahl, H. W. Grießhammer, J. A. McGovern, D. R. Phillips, and M. T. Pratola. Designing optimal experiments: an application to proton compton scattering. *The European Physical Journal A*, 57(3), February 2021. ISSN 1434-601X. doi: 10.1140/epja/s10050-021-00382-2. URL http://dx.doi.org/10.1140/epja/s10050-021-00382-2.

[104] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129), 1978.

[105] Henry Moss, David Leslie, Daniel Beck, Javier González, and Paul Rayson. Boss: Bayesian optimization over string spaces. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15476–15486. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/b19aa25ff58940d974234b48391b9549-Paper.pdf.

[106] Henry B Moss, David S Leslie, and Paul Rayson. Mumbo: Multi-task max-value bayesian optimization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 447–462. Springer, 2020.

[107] Henry B. Moss, David S. Leslie, Javier Gonzalez, and Paul Rayson. Gibbon: General-purpose information-based bayesian optimisation. *Journal of Machine Learning Research*, 22(235):1–49, 2021. URL http://jmlr.org/papers/v22/21-0120.html.

[108] Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KSugKcbNf9.

[109] Samuel Müller, Matthias Feurer, Noah Hollmann, and Frank Hutter. PFNs4BO: In-context learning for Bayesian optimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 25444–25470. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/muller23a.html.

[110] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL probml.ai.

[111] I. Murray and R. Adams. Slice sampling covariance hyperparameters of latent Gaussian models. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Proceedings of the 24th International Conference on Advances in Neural Information Processing Systems (NeurIPS'10)*, pages 1732–1740, 2010.

[112] Iain Murray, Ryan Adams, and David MacKay. Elliptical slice sampling. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 541–548, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL `https://proceedings.mlr.press/v9/murray10a.html`.

[113] Mojmir Mutny and Andreas Krause. Efficient high dimensional bayesian optimization with additivity and quadrature fourier features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/4e5046fc8d6a97d18a5f54beaed54dea-Paper.pdf`.

[114] L. Nardi, D. Koeplinger, and K. Olukotun. Practical design space exploration. In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 347–358. IEEE, 2019.

[115] Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*, pages 4752–4761. PMLR, 2019.

[116] Radford M. Neal. Mcmc using hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, 1996.

[117] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 1996.

[118] Willie Neiswanger, Ke Alexander Wang, and Stefano Ermon. Bayesian algorithm execution: Estimating computable properties of black-box functions using mutual information. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8005–8015. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/neiswanger21a.html`.

[119] Willie Neiswanger, Lantao Yu, Shengjia Zhao, Chenlin Meng, and Stefano Ermon. Generalizing bayesian optimization with decision-theoretic entropies. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713871088.

[120] Vu Nguyen and Michael A. Osborne. Knowing the what but not the where in Bayesian optimization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7317–7326. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/nguyen20d.html`.

[121] C. Oh, E. Gavves, and M. Welling. BOCK : Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*, pages 3865–3874, 2018.

[122] Changyong Oh, Jakub M. Tomczak, Efstratios Gavves, and Max Welling. *Combinatorial Bayesian optimization using the graph cartesian product*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[123] Michael A Osborne. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, Oxford University, UK, 2010.

[124] C. Papadimitriou. Optimal sensor placement methodology for parametric identification of structural systems. *Journal of Sound and Vibration*, 278 (4):923–947, 2004. ISSN 0022-460X. doi: https://doi.org/10.1016/j.jsv.2003.10.063. URL `https://www.sciencedirect.com/science/article/pii/S0022460X04000355`.

[125] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Increasing the scope as you learn: Adaptive bayesian optimization in nested subspaces. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=e4Wf6112DI`.

[126] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Bounce: a Reliable Bayesian Optimization Algorithm for Combinatorial and Mixed Spaces. In *Advances in Neural Information Processing Systems*, 2023.

[127] V. Perrone, R. Jenatton, M. Seeger, and C. Archambeau. Scalable hyperparameter transfer learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Proceedings of*

the 32nd International Conference on Advances in Neural Information Processing Systems (NeurIPS'18), pages 12751–12761, 2018.

[128] Chao Qin, Diego Klabjan, and Daniel Russo. Improving the expected improvement algorithm. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5387–5397, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[129] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, Advances in Neural Information Processing Systems, page 1177–1184, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520.

[130] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL `https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf`.

[131] Tom Rainforth, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. Modern bayesian experimental design, 2023.

[132] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[133] Johann Moritz Reumschüssel, Jakob Georg Raimund von Saldern, Bernhard Ćosić, and Christian Oliver Paschereit. Data-driven optimization of a gas turbine combustor: A bayesian approach addressing no x emissions, lean extinction limits, and thermoacoustic stability. *Data-Centric Engineering*, 2024. URL `https://api.semanticscholar.org/CorpusID:274158611`.

[134] Christoffer Riis, Francisco N Antunes, Frederik Boe Hüttel, Carlos Lima Azevedo, and Francisco Camara Pereira. Bayesian active learning with fully bayesian gaussian processes. *arXiv preprint arXiv:2205.10186*, 2022.

[135] Binxin Ru, Michael A. Osborne, Mark Mcleod, and Diego Granziol. Fast information-theoretic Bayesian optimisation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4384–4392. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/ru18a.html`.

[136] Binxin Ru, Xingchen Wan, Xiaowen Dong, and Michael Osborne. Interpretable neural architecture search via bayesian optimisation with weisfeiler-lehman kernels. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=j9Rv7qdXjd.

[137] Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *Journal of Machine Learning Research*, 17(68):1–30, 2016. URL http://jmlr.org/papers/v17/14-087.html.

[138] Daniel J. Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018. ISSN 1935-8237. doi: 10.1561/2200000070. URL http://dx.doi.org/10.1561/2200000070.

[139] Elizabeth G. Ryan, Christopher C. Drovandi, James M. McGree, and Anthony N. Pettitt. A review of modern computational algorithms for bayesian optimal design. *International Statistical Review / Revue Internationale de Statistique*, 84(1):128–154, 2016. ISSN 03067734, 17515823. URL http://www.jstor.org/stable/44162464.

[140] Elizabeth G Ryan, Christopher C Drovandi, James M McGree, and Anthony N Pettitt. A review of modern computational algorithms for bayesian optimal design. *International Statistical Review*, 84(1):128–154, 2016.

[141] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. ISBN 0262194759.

[142] Paola Sebastiani and Henry P. Wynn. Maximum entropy sampling and optimal bayesian experimental design. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 62(1):145–157, 2000. ISSN 13697412, 14679868. URL http://www.jstor.org/stable/2680683.

[143] Matthias Seeger. Gaussian processes for machine learning. *International Journal of Neural Systems*, 14(02):69–106, 2004. doi: 10.1142/S0129065704001899. URL https://doi.org/10.1142/S0129065704001899. PMID: 15112367.

[144] B. Shahriari, K. Swersky, Z. Wang, R. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[145] L. Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[146] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14:199–222, 2004.

[147] J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian optimization of machine learning algorithms. In Bartlett et al. [9], pages 2960–2968.

[148] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, page 2951–2959, Red Hook, NY, USA, 2012. Curran Associates Inc.

[149] J. Snoek, K. Swersky, R. Zemel, and R. Adams. Input warping for Bayesian optimization of non-stationary functions. In Xing and Jebara [179], pages 1674–1682.

[150] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, Prabhat, and R. Adams. Scalable Bayesian optimization using deep neural networks. In Bach and Blei [5], pages 2171–2180.

[151] Jiaming Song, Lantao Yu, Willie Neiswanger, and Stefano Ermon. A general recipe for likelihood-free Bayesian optimization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20384–20404. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/song22b.html`.

[152] Xingyou Song, Qiuyi Zhang, Chansoo Lee, Emily Fertig, Tzu-Kuo Huang, Lior Belenki, Greg Kochanski, Setareh Ariafar, Srinivas Vasudevan, Sagi Perel, and Daniel Golovin. The vizier gaussian process bandit algorithm. *Google DeepMind Technical Report*, 2024.

[153] A. Souza, L. Nardi, L. Oliveira, K. Olukotun, M. Lindauer, and F. Hutter. Bayesian optimization with a prior for the optimum. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part III*, volume 12977 of *Lecture Notes in Computer Science*, pages 265–296. Springer, 2021.

[154] J. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust Bayesian neural networks. In D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Proceedings of the 30th International Conference on Advances in Neural Information Processing Systems (NeurIPS'16)*, 2016.

[155] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*, pages 1015–1022. Omnipress, 2010.

[156] Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20459–20478. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/stanton22a.html`.

[157] Jonathan Styrud, Matthias Mayr, Erik Hellsten, Volker Krueger, and Christian Smith. Bebop - combining reactive planning and bayesian optimization to solve robotic manipulation tasks. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16459–16466, 2024. doi: 10.1109/ICRA57147.2024.10611468.

[158] K. J. Swersky. *Improving Bayesian Optimization for Machine Learning using Expert Priors*. PhD thesis, University of Toronto, 2017.

[159] Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9334–9345. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/takeno20a.html`.

[160] Max Tegmark, Daniel J. Eisenstein, Michael A. Strauss, David H. Weinberg, Michael R. Blanton, Joshua A. Frieman, Masataka Fukugita, James E. Gunn, Andrew J. S. Hamilton, Gillian R. Knapp, Robert C. Nichol, Jeremiah P. Ostriker, Nikhil Padmanabhan, Will J. Percival, David J. Schlegel, Donald P. Schneider, Roman Scoccimarro, Uroš Seljak, Hee-Jong Seo, Molly Swanson, Alexander S. Szalay, Michael S. Vogeley, Jaiyul Yoo, Idit Zehavi, Kevork Abazajian, Scott F. Anderson, James Annis, Neta A. Bahcall, Bruce Bassett, Andreas Berlind, Jon Brinkmann, Tamás Budavari, Francisco Castander, Andrew Connolly, Istvan Csabai, Mamoru Doi, Douglas P. Finkbeiner, Bruce Gillespie, Karl Glazebrook, Gregory S. Hennessy, David W. Hogg, Željko Ivezić, Bhuvnesh Jain, David Johnston,

Stephen Kent, Donald Q. Lamb, Brian C. Lee, Huan Lin, Jon Loveday, Robert H. Lupton, Jeffrey A. Munn, Kaike Pan, Changbom Park, John Peoples, Jeffrey R. Pier, Adrian Pope, Michael Richmond, Constance Rockosi, Ryan Scranton, Ravi K. Sheth, Albert Stebbins, Christopher Stoughton, István Szapudi, Douglas L. Tucker, Daniel E. vanden Berk, Brian Yanny, and Donald G. York. Cosmological constraints from the SDSS luminous red galaxies. Phys. Rev. D, 74(12):123507, December 2006. doi: 10.1103/PhysRevD.74.123507.

[161] W. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4): 285–294, 1933.

[162] Louis C Tiao, Aaron Klein, Matthias W Seeger, Edwin V. Bonilla, Cedric Archambeau, and Fabio Ramos. Bore: Bayesian optimization by density-ratio estimation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10289–10300. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/tiao21a.html`.

[163] Dustin Tran, Rajesh Ranganath, and David M Blei. The variational gaussian process. *International Conference on Learning Representations*, 2016.

[164] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[165] Julien Villemonteix, Emmanuel Vazquez, and Eric Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44, 12 2006. doi: 10.1007/s10898-008-9354-2.

[166] Q. Wang, Y. Ming, Z. Jin, Q. Shen, D. Liu, M. J. Smith, K. Veeramachaneni, and H. Qu. Atmseer: Increasing transparency and controllability in automated machine learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12. Association for Computing Machinery, 2019.

[167] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

[168] Z. Wang, C. Li, S. Jegelka, and P. Kohli. Batched High-dimensional Bayesian Optimization via Structural Kernel Learning. In D. Precup and Y. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, volume 70, pages 3656–3664. Proceedings of Machine Learning Research, 2017.

[169] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning (ICML)*, 2017.

[170] Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 745–754. PMLR, 09–11 Apr 2018. URL `https://proceedings.mlr.press/v84/wang18c.html`.

[171] Ziyu Wang and Nando de Freitas. Theoretical analysis of bayesian optimisation with unknown gaussian process hyper-parameters, 2014.

[172] Christopher Williams and Carl Rasmussen. Gaussian processes for regression. In D. Touretzky, M.C. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995. URL `https://proceedings.neurips.cc/paper_files/paper/1995/file/7cce53cf90577442771720a370c3c723-Paper.pdf`.

[173] Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1067–1075, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL `https://proceedings.mlr.press/v28/wilson13.html`.

[174] James Wilson, Frank Hutter, and Marc Deisenroth. Maximizing acquisition functions for bayesian optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper/2018/file/498f2c21688f6451d9f5fd09d53edda7-Paper.pdf`.

[175] James T. Wilson. Stopping bayesian optimization with probabilistic regret bounds, 2024.

[176] James T. Wilson, Riccardo Moriconi, Frank Hutter, and Marc Peter Deisenroth. The reparameterization trick for acquisition functions, 2017. URL https://arxiv.org/abs/1712.00424.

[177] James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Efficiently sampling functions from gaussian process posteriors. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/2002.09309.

[178] Jian Wu, Matthias Poloczek, Andrew G Wilson, and Peter Frazier. Bayesian optimization with gradients. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/64a08e5f1e6c39faeb90108c430eb120-Paper.pdf.

[179] E. Xing and T. Jebara, editors. *Proceedings of the 31th International Conference on Machine Learning, (ICML'14)*, 2014. Omnipress.

[180] Juliusz Ziomek, Masaki Adachi, and Michael A. Osborne. Bayesian optimisation with unknown hyperparameters: Regret bounds logarithmically closer to optimal, 2024. URL https://arxiv.org/abs/2410.10384.

[181] Juliusz Krzysztof Ziomek and Haitham Bou Ammar. Are random decompositions all we need in high dimensional Bayesian optimisation? In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 43347–43368. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ziomek23a.html.

Chapter 1

Scientific publications

Author contributions

Tab. 1.1: Overview of contributions in each paper included in the thesis.

Paper	Concept	Implementation	Evaluation	Writing
I	◑	◕	◕	◑
II	●	●	●	◕
III	◕	●	●	◕
IV	●	●	●	◕
V	◑	◔	◔	◔
VI	◕	●	●	◕

The dark portion of the circle represents the amount of work and responsibilities assigned to Carl Hvarfner for each individual step.

- ◔ Carl Hvarfner was a minor contributor to the work
- ◑ Carl Hvarfner was a contributor to the work
- ◕ Carl Hvarfner did most of the work
- ● Carl Hvarfner did almost all of the work

Co-authors are abbreviated as follows:
CH - Carl Hvarfner
LN - Luigi Nardi
FH - Frank Hutter
EH - Erik Hellsten
LP - Leonard Papenmeier
DS - Danny Stoll
AS - Artur Souza
ML - Marius Lindauer

Paper I CH, AS, FH, LN and ML designed the method. CH coded the algorithm, developed the theory and ran the majority of experiments. DS set up and ran the DNN experiments. CH, DS, FH and LN primarily wrote the paper, with all authors participating. All figures were produced by CH. FH and LN contributed the experimental setup.

Paper II CH designed the method through discussions with FH and LN. CH coded the algorithm, ran experiments and produced figures for the paper. CH produced a majority of the writing, with assistance from FH and LN. All figures were produced by CH. FH provided the computational resources.

Paper III CH designed the method with input from EH, FH and LN. CH coded the algorithm, ran experiments and produced figures for the paper. FH and LN helped with writing, all figures were produced by CH. FH provided the experimental setup.

Paper IV CH designed the concept, coded the algorithm and ran all experiments in the paper. FH and LN assisted in algorithm development. LN and FH contributed in writing the paper. FH contributed in experimental setup.

Paper V CH proposed the main concept. LP and EH designed the method and coded the algorithm, with input from CH. LP and EH primarily wrote the paper, with contributions from CH and LN.

Paper VI CH designed the method and coded the algorithm. EH primarily developed the theoretical pieces, with input from CH. EH and LN participated in discussions and assisted in writing the paper.

Paper I

πBO: Augmenting Acquisition Functions with User Beliefs for Bayesian Optimization

Carl Hvarfner
Lund University

Danny Stoll
University of Freiburg

Artur Souza
Federal University of Minas Gerais

Marius Lindauer
Leibniz University Hannover

Frank Hutter
University of Freiburg

Luigi Nardi
Lund University

Abstract

Bayesian optimization has become an established framework and popular tool for hyperparameter optimization (HPO) of machine learning (ML) algorithms. While known for its sample-efficiency, vanilla BO can not utilize readily available prior beliefs the practitioner has on the potential location of the optimum. Thus, BO disregards a valuable source of information, reducing its appeal to ML practitioners. To address this issue, we propose πBO, an acquisition function generalization which incorporates prior beliefs about the location of the optimum in the form of a probability distribution, provided by the user. In contrast to previous approaches, πBO is conceptually simple and can easily be integrated with existing libraries and many acquisition functions. We provide regret bounds when πBO is applied to the common Expected Improvement acquisition function and prove convergence at regular rates independently of the prior. Further, our experiments show that πBO outperforms competing approaches across a wide suite of benchmarks and prior characteristics. We also demonstrate that πBO improves on the state-of-the-art performance for a popular deep learning task, with a 12.5$\times$ time-to-accuracy speedup over prominent BO approaches.

1 Introduction

The optimization of expensive black-box functions is a prominent task, arising across a wide range of applications. Bayesian optimization (BO) is a sample-efficient approach to cope with this task, and has been successfully applied to various problem settings, including hyperparameter optimization (HPO) [45], neural architecture search (NAS) [40], joint NAS and HPO [61], algorithm configuration [20], hardware design [33], robotics [9], and the game of Go [10].

Despite the demonstrated effectiveness of BO for HPO [5, 54], its adoption among practitioners remains limited. In a survey covering NeurIPS 2019 and ICLR 2020 [6], manual search was shown to be the most prevalent tuning method, with BO accounting for less than 7% of all tuning efforts. As the understanding of hyperparameter settings in deep learning (DL) models increase [44], so too does the tuning proficiency of practitioners [1]. As previously displayed [44, 1, 47, 58], this knowledge manifests in choosing single configurations or regions of hyperparameters that presumably yield good results, demonstrating a belief over the location of the optimum. BO's deficit to properly incorporate said beliefs is a reason why practitioners prefer manual search to BO [58], despite its documented shortcomings [4]. To improve the usefulness of automated HPO approaches for ML practictioners, the ability to incorporate such knowledge is pivotal.

Well-established BO frameworks [45, 20, 16, 23, 3] support user input to a limited extent, such as by biasing the initial design, or by narrowing the search space; however, this type of hard prior can lead to poor performance by missing important regions. BO also supports a prior over functions $p(f)$ via the Gaussian Process kernel. However, this option for injecting knowledge is not aligned with the knowledge that experts possess: they often know which *ranges of hyperparameter values* tend to work best [35, 44, 58], and are able to specify a probability distribution to quantify these priors. For example, many users of the Adam optimizer [24] know that its best learning rate is often in the vicinity of 1×10^{-3}. In practice, DL experiments are typically conducted in a low-budget setting of less than 50 full model trainings [6]. As such, practitioners want to exploit their knowledge efficiently without wasting early model trainings on configurations they expect to likely perform poorly. Unfortunately, this suits standard BO poorly, as BO requires a moderate number of function evaluations to learn about the response surface and make informed decisions that outperform random search.

While there is a demand to increase knowledge injection possibilities to further the adoption of BO, the concept of encoding prior beliefs over the location of

an optimum is still rather novel: while there are some initial works [37, 29, 47], no approach exists so far that allows the integration of arbitrary priors and offers flexibility in the choice of acquisition function; theory is also lacking. We close this gap by introducing a novel, remarkably simple, approach for injecting arbitrary prior beliefs into BO that is easy to implement, agnostic to the surrogate model used and converges at standard BO rates for any choice of prior.

Our contributions After discussing our problem setting, related work, and background (Section 2), we make the following contributions:

1. We introduce πBO, a novel generalization of myopic acquisition functions that accounts for user-specified prior distributions over possible optima, is demonstrably simple-to-implement, and can be easily combined with arbitrary surrogate models (Section 3.1 & 3.2);
2. We formally prove that πBO inherits the theoretical properties of the well-established Expected Improvement acquisition function (Section 3.3);
3. We demonstrate on a broad range of established benchmarks and in DL case studies that πBO can yield 12.5× time-to-accuracy speedup over vanilla BO (Section 4).

2 Background and Related Work

2.1 Black-box Optimization

We consider the problem of optimizing a black-box function f across a set of feasible inputs $\mathcal{X} \subset \mathbb{R}^d$:

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}). \tag{1.1}$$

We assume that $f(x)$ is expensive to evaluate, and can potentially only be observed through a noisy estimate, y. In this setting, we wish to minimize f in an efficient manner, typically adhering to a budget which sets a cap on the number of points that can be evaluated.

Black-Box Optimization with Probabilistic User Beliefs In our work, we consider an augmented version of the optimization problem in Eq. (1.1), where we have access to user beliefs in the form of a probability distribution

on the location of the optimum. Formally, we define the problem of black-box optimization with probabilistic user beliefs as solving Eq. (1.1), given a user-specified prior probability on the location of the optimum defined as

$$\pi(\boldsymbol{x}) = \mathbb{P}\left(f(\boldsymbol{x}) = \min_{\boldsymbol{x}' \in \mathcal{X}} f(\boldsymbol{x}')\right), \tag{1.2}$$

where regions that the user expects to likely to contain an optimum will have a high value. We note that, without loss of generality, we require π to be strictly positive on all of $\mathcal{X}$, i.e., any point in the search space might be an optimum. Since the user belief $\pi(\boldsymbol{x})$ can be inaccurate or even misleading, optimizing Eq. (1.1) given (1.2) is a challenging problem.

2.2 Bayesian Optimization

We outline Bayesian optimization [32, 7, 42].

Model BO aims to globally minimize f by an initial experimental design $\mathcal{D}_0 = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{M}$ and thereafter sequentially deciding on new points $\boldsymbol{x}_n$ to form the data $\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{(\boldsymbol{x}_n, y_n)\}$ for the n-th iteration with $n \in \{1 \dots N\}$. After each new observation, BO constructs a probabilistic surrogate model of f and uses that surrogate to evaluate an acquisition function $\alpha(\boldsymbol{x}, \mathcal{D}_n)$. The combination of surrogate model and acquisition function encodes the policy for selecting the next point $\boldsymbol{x}_{n+1}$. When constructing the surrogate, the most common choice is Gaussian processes [38], which model f as $p(f|\mathcal{D}_n) = \mathcal{GP}(m, k)$, with prior mean m (which is typically 0) and positive semi-definite covariance kernel k. The posterior mean m_n and the variance s_n^2 are

$$m_n(\boldsymbol{x}) = \mathbf{k}_n(\boldsymbol{x})^\top (\mathbf{K}_n + \sigma_n^2 \mathbf{I})\mathbf{y} \tag{1.3}$$

$$s_n^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \mathbf{k}_n(\boldsymbol{x})^\top (\mathbf{K}_n + \sigma_n^2 \mathbf{I})\mathbf{k}_n(\boldsymbol{x}), \tag{1.4}$$

where $(\mathbf{K}_n)_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\mathbf{k}_n(\boldsymbol{x}) = [k(\boldsymbol{x}, \boldsymbol{x}_1), \dots, k(\boldsymbol{x}, \boldsymbol{x}_n)]^\top$ and σ_n^2 is the estimation of the observation noise variance σ^2. Alternative surrogate models include Random forests [20] and Bayesian neural networks [48].

Acquisition Functions To obtain new candidates to evaluate, BO employs a criterion, called an acquisition function, that encapsulates an explore-exploit trade-off. By maximizing this criterion at each iteration, one or more candidate point are obtained and added to observed data. Several acquisition functions

are used in BO; the most common of these is Expected Improvement (EI) [21]. For a noiseless function, EI selects the next point $\boldsymbol{x}_{n+1}$, where f_n^* is the minimal objective function value observed by iteration n, as

$$\boldsymbol{x}_{n+1} \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}\left[[(f_n^* - f(\boldsymbol{x})]^+\right] = \arg\max_{\boldsymbol{x}\in\mathcal{X}} Z s_n(\boldsymbol{x})\Phi(Z) + s_n(\boldsymbol{x})\phi(Z), \quad (1.5)$$

where $Z = (f_n^* - m_n(\boldsymbol{x}))/s_n(\boldsymbol{x})$. Thus, EI provides a myopic strategy for determining promising points; it also comes with convergence guarantees [8]. Similar myopic acquisition functions are Upper Confidence Bound (UCB) [49], Probability of Improvement (PI) [22, 26] and Thompson Sampling (TS) [53]. A different class of acquisition functions is based on non-myopic criteria, such as Entropy Search [17], Predictive Entropy Search [18] and Max-value Entropy Search [59], which select points to minimize the uncertainty about the optimum, and the Knowledge Gradient [15], which aims to minimize the posterior mean of the surrogate at the subsequent iteration. Our work applies to all acquisition functions in the first class, and we leave its extension to those in the second class for future work.

2.3 Related Work

There are two main categories of approaches that exploit prior knowledge in BO: approaches that use records of previous experiments, and approaches that incorporate assumptions on the black-box function provided either directly or indirectly by the user. As πBO exploits prior knowledge from users, we briefly discuss approaches which utilize previous experiments, and then comprehensively discuss the literature on exploiting expert knowledge.

Learning from Previous Experiments Transfer learning for BO aims to automatically extract and use knowledge from prior executions of BO. These executions can come, for example, from learning and optimizing the hyperparameters of a machine learning algorithm on different datasets [55, 51, 60, 35, 13, 14], or from optimizing the hyperparameters at different development stages [50]. For a comprehensive overview of meta learning for hyperparameter optimization, please see the survey from [56]. In contrast to these transfer learning approaches, πBO and the related work discussed below does not hinge on the existence of previous experiments, and can therefore be applied more generally.

Incorporating Expert Priors over Function Structure BO can leverage structural priors on how the objective function is expected to behave. Traditionally, this is done via the surrogate model's prior over functions, e.g., the kernel of the GP. However, there are lines of work that explore additional structural

priors for BO to leverage. For instance, both SMAC [20] and iRace [30] support structural priors in the form of log-transformations, [28] propose to use knowledge about the monotonicity of the objective function as a prior for BO, and [46] model non-stationary covariance between inputs by warping said inputs.

[34] and [43] both propose structural priors tailored to high-dimensional problems, addressing the issue of over-exploring the boundary described by [52]. [34] propose a cylindrical kernel that expands the center of the search space and shrinks the edges, while [43] propose adding derivative signs to the edges of the search space to steer BO towards the center. Lastly, [41] propose a BO algorithm for unbounded search spaces which uses a regularizer to penalize points based on their distance to the center of the user-defined search space. All of these approaches incorporate prior information on specific properties of the function or search space, and are thus not always applicable. Moreover, they do not generally direct the search to desired regions of the search space, offering the user little control over the selection of points to evaluate.

Incorporating Expert Priors over Function Optimum Few previous works have proposed to inject explicit prior distributions over the location of an optimum into BO. In these cases, users explicitly define a prior that encodes their beliefs on where the optimum is more likely to be located. [5] suggest an approach that supports prior beliefs from a fixed set of distributions. However, this approach cannot be combined with standard acquisition functions. BOPrO [47] employs a similar structure that combines the user-provided prior distribution with a data-driven model into a pseudo-posterior. From the pseudo-posterior, configurations are selected using the EI acquisition function, using the formulation in [5]. While BOPrO is able to recover from misleading priors, its design restricts it to only use EI. Moreover, it does not provide the convergence guarantees of πBO.

[29] propose to infer a posterior conditioned on both the observed data and the user prior through repeated Thompson sampling and maximization under the prior. This method displays robustness against misleading priors but lacks in empirical performance. Additionally, it is restricted to only one specific acquisition function. [37] use the probability integral transform to warp the search space, stretching high-probability regions and shrinking others. While the approach is model- and acquisition function agnostic, it requires invertible priors, and does not empirically display the ability to recover from misleading priors. In Section 4, we demonstrate that πBO compares favorably for priors over the function optimum, and shows improved empirical performance.

In summary, πBO sets itself apart from the methods above by being simpler

(and thus easier to implement in different frameworks), flexible with regard to different acquisition functions and different surrogate models, the availability of theoretical guarantees, and, as we demonstrate in Section 4, better empirical results.

3 Methodology

We now present πBO, which allows users to specify their belief about the location of the optimum through any probability distribution. A conceptually simple approach, πBO can be easily implemented in existing BO frameworks and can be combined directly with the myopic acquisition functions listed above. πBO augments an acquisition function to emphasize promising regions under the prior, ensuring such regions are to be explored frequently. As optimization progresses, the πBO strategy increasingly resembles that of vanilla BO, retaining its standard convergence rates (see Section 3.3). πBO is publicly available as part of the SMAC (`https://github.com/automl/SMAC3`) and HyperMapper (`https://github.com/luinardi/hypermapper`) HPO frameworks.

3.1 Prior-weighted Acquisition Function

In πBO, we consider $\pi(\boldsymbol{x})$ in Eq. (1.2) to be a weighting scheme on points in $\mathcal{X}$. The heuristic provided by an acquisition function $\alpha(\boldsymbol{x}, \mathcal{D}_n)$, such as EI in Eq. (2.2), can then be combined with said weighting scheme to form a prior-weighted version of the acquisition function. The resulting strategy then becomes:

$$\boldsymbol{x}_n \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha(\boldsymbol{x}, \mathcal{D}_n)\pi(\boldsymbol{x}). \tag{1.6}$$

This emphasizes good points under $\pi(\boldsymbol{x})$ throughout the optimization. While this property is suitable for well-located priors π, it risks incurring a substantial slowdown for poorly-chosen priors; we will now show how to counter this by decaying the prior over time.

3.2 Decaying Prior-weighted Acquisition Function

As the optimization progresses, we should increasingly trust the surrogate model over the prior; the model improves with data while the user prior remains fixed. This cannot be achieved with the formulation in Eq. (1.6), as poorly-chosen priors would permanently slow down the optimization. Rather, to accomplish this

desired behaviour, the influence of the prior needs to decay over time. Building on the approaches of [27] and [47], we accomplish this by raising the prior to a power $\gamma_n \in \mathbb{R}^+$, which decays towards zero with growing n. Thus, the resulting prior $\pi_n(\boldsymbol{x}) = \pi(\boldsymbol{x})^{\gamma_n}$ reflects a belief on the location of an optimum that gets weaker with time, converging towards a uniform distribution. We set $\gamma_n = \beta/n$, where $\beta \in \mathbb{R}^+$ is a hyperparameter set by the user, reflecting their confidence in $\pi(\boldsymbol{x})$. For a given acquisition function $\alpha(\boldsymbol{x}, \mathcal{D}_n)$ and user-specified prior $\pi(\boldsymbol{x})$, we define the decaying prior-weighted acquisition function at iteration n as

$$\alpha_{\pi,n}(\boldsymbol{x}, \mathcal{D}_n) \triangleq \alpha(\boldsymbol{x}, \mathcal{D}_n)\pi_n(\boldsymbol{x}) \triangleq \alpha(\boldsymbol{x}, \mathcal{D}_n)\pi(\boldsymbol{x})^{\beta/n} \tag{1.7}$$

and its accompanying strategy as the maximizer of $\alpha_{\pi,n}$. With the acquisition function in Eq. (1.7), the prior will assume large importance initially, promoting the selection of points close to the prior mode. With time, the exponent on the prior will tend to zero, making the prior tend to uniform. Thus, with increasing n, the point selection of $\alpha_{\pi,n}$ becomes increasingly similar to that of α. Algorithm 1.1 displays the simplicity of the new strategy, highlighting the required one-line change (Line 6) in the main BO loop. In Line 3, the mode of the prior is used as a first initial sample if available. Otherwise, only sampling is used for initialization.

Algorithm 1.1 πBO Algorithm

1: **Input:** Input space $\mathcal{X}$, prior distribution over optimum $\pi(\boldsymbol{x})$, prior confidence parameter β, size M of the initial design, max number of optimization iterations N.
2: **Output:** Optimized design $\boldsymbol{x}^*$.
3: $\{\boldsymbol{x}_i\}_{i=1}^M \sim \pi(\boldsymbol{x})$, $\{y_i \leftarrow f(\boldsymbol{x}_i) + \varepsilon_i\}_{i=1}^M, \quad \varepsilon_i \sim N(0, \sigma^2)$
4: $\mathcal{D}_0 \leftarrow \{(\boldsymbol{x}_i, y_i)\}_{i=1}^M$
5: **for** $\{n = 1, 2, \ldots, N\}$ **do**
6: $\quad \boldsymbol{x}_{new} \leftarrow \arg\max_{\boldsymbol{x} \in \mathcal{X}} \alpha(\boldsymbol{x}, \mathcal{D}_{n-1})\pi(\boldsymbol{x})^{\beta/n}$
7: $\quad y_{new} \leftarrow f(\boldsymbol{x}_{new}) + \varepsilon_i$
8: $\quad \mathcal{D}_n \leftarrow \mathcal{D}_{n-1} \cup \{(\boldsymbol{x}_{new}, y_{new})\}$
9: **end for**
10: **return** $\boldsymbol{x}^* \leftarrow \arg\min_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_N} y_i$

To illustrate the behaviour of πBO, we consider a toy problem with Gaussian priors on three different locations of the 1D space (center, left and right) as displayed in Figure 1.1. We define a 1D-Log-Branin toy problem by setting the second dimension of the 2D Branin function to the global optimum $x_2 = 2.275$ and optimizing for the first dimension. Initially (iteration 4 in the top row),

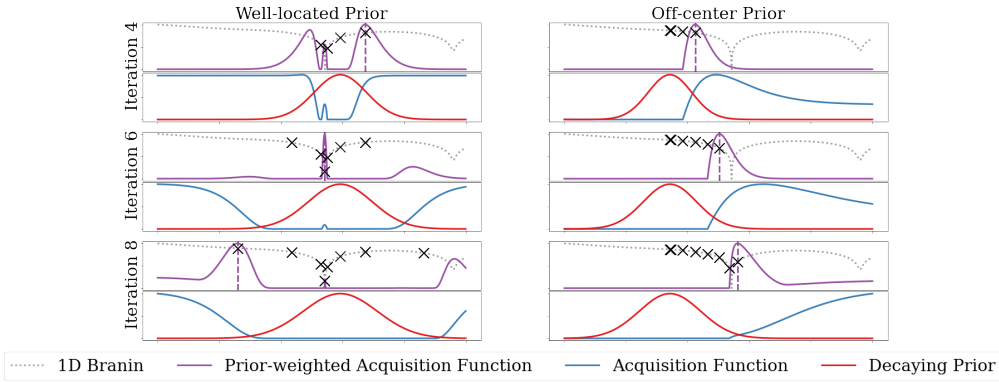

Fig. 1.1: Rescaled values of prior-weighted EI (purple), EI (blue) and π_n (red) on a 1D-Branin in logscale (grey) with global optimum in the center of the search space. Runs with two different prior locations ("Well-located" slightly right of optimum, "Off-center" significantly left of optimum) are shown in the two columns. Each row represents an iteration (iteration 4, 6 and 8), for an optimization run with $\beta = 2$. The current selection can be seen as a vertical violet line, and all previous observations are marked as crosses. πBO amplifies EI in a gradually increasing region around the prior, and moves away from the prior as iterations progress. This is particularly evident in the Off-center example.

πBO amplifies the acquisition function α in high-probability regions, putting a lot of trust in the prior. As the prior decays (iteration 6 and 8 in the middle and bottom rows, respectively), the influence of the prior on the point selection decreases. By later iterations, πBO has searched substantially around the prior mode, and moves gradually towards other parts of the search space. This is of particular importance for the scenarios in the right column, where πBO recovers from a misleading prior.

3.3 Theoretical Analysis

We now study the πBO method from a theoretical standpoint when paired with the EI acquisition function. To provide convergence rates, we rely on the set of assumptions introduced by [8]. These assumptions are satisfied for popular kernels like the [31] class and the Gaussian kernel, which is obtained in the limit $\nu \to \infty$, where the rate ν controls the smoothness of functions from the GP prior. Our theoretical results apply when both length scales $\boldsymbol{\ell}$ and the global scale of variation σ are fixed; these results can then be extended to the case where the kernel hyperparameters are learned using Maximum Likelihood Estimation (MLE) following the same procedure as in [8] (Theorem 5). We define the loss over the ball B_R for a function f of norm $||f||_{\mathcal{H}_{\boldsymbol{\ell}}(\mathcal{X})} \leqslant R$ in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}_{\boldsymbol{\ell}}(\mathcal{X})$ given a symmetric positive-definite kernel

$K_{\boldsymbol{\ell}}$ as

$$\mathcal{L}_n(u, \mathcal{D}_n, \mathcal{H}_{\boldsymbol{\ell}}(\mathcal{X}), R) \triangleq \sup_{||f||_{\mathcal{H}_{\boldsymbol{\ell}}(\mathcal{X})} \leqslant R} \mathbb{E}_f^u[f(\boldsymbol{x}_n^*) - \min f], \tag{1.8}$$

where n is the optimization iteration and u a strategy. We focus on the strategy that maximizes EI_π, the prior-weighted EI, and show that the loss in Equation (1.8) can, at any iteration n, be bounded by the vanilla EI loss function. We refer to $\mathrm{EI}_{\pi,n}$ and EI_n when we want to emphasize the iteration n for the acquisition functions EI_π and EI, respectively.

Theorem 1. *Given $\mathcal{D}_n$, $K_{\boldsymbol{\ell}}$, π, β, σ, $\boldsymbol{\ell}$, R and the compact set $\mathcal{X} \subset \mathbb{R}^d$ as defined above, the loss $\mathcal{L}_n$ incurred at iteration n by $EI_{\pi,n}$ can be bounded from above as*

$$\mathcal{L}_n(EI_{\pi,n}, \mathcal{D}_n, \mathcal{H}_{\boldsymbol{\ell}}(\mathcal{X}), R) \leqslant C_{\pi,n} \mathcal{L}_n(EI_n, \mathcal{D}_n, \mathcal{H}_{\boldsymbol{\ell}}(\mathcal{X}), R), \tag{1.9}$$

where

$$C_{\pi,n} = \left(\frac{\max_{\boldsymbol{x} \in \mathcal{X}} \pi(\boldsymbol{x})}{\min_{\boldsymbol{x} \in \mathcal{X}} \pi(\boldsymbol{x})} \right)^{\beta/n}. \tag{1.10}$$

Using Theorem 1, we obtain the convergence rate of EI_π. This trivially follows when considering the fraction of the losses in the limit and inserting the original convergence rate on EI as in [8]:

Corollary 1. *The loss of a decaying prior-weighted Expected Improvement strategy, EI_π, is asymptotically equal to the loss of an Expected Improvement strategy, EI:*

$$\mathcal{L}_n(EI_{\pi,n}, \mathcal{D}_n, \mathcal{H}_{\boldsymbol{\ell}}(\mathcal{X}), R) \sim \mathcal{L}_n(EI_n, \mathcal{D}_n, \mathcal{H}_{\boldsymbol{\ell}}(\mathcal{X}), R), \tag{1.11}$$

so we obtain a convergence rate for EI_π of

$$\mathcal{L}_n(EI_{\pi,n}, \mathcal{D}_n, \mathcal{H}_{\boldsymbol{\ell}}(\mathcal{X}), R) = \mathcal{O}(n^{-(\nu \wedge 1)/d} (\log n)^\gamma). \tag{1.12}$$

Thus, we determine that the weighting introduced by EI_π does not negatively impact the worst-case convergence rate. The short-term performance is controlled by the user in their choice of $\pi(\boldsymbol{x})$ and β. This result is coherent with intuition, as a weaker prior or quicker decay will yield a short-term performance closer to that of EI. In contrast, a stronger prior or slower decay does not guarantee the same short-term performance, but can produce better empirical results, as shown in Section 4.

4 Results

We empirically demonstrate the efficiency of πBO in three different settings. As πBO is a general method to augment acquisition functions, it can be implemented in different parent BO packages, and the implementation in any given package inherits the pros and cons of that package. To minimize confounding factors concerning this choice of parent package, we keep comparisons within the methods in one package where possible. In Sec. 4.2, using Spearmint as a parent package, we evaluate πBO against three intuitive baselines to assess its performance and robustness on priors with different qualities, ranging from very accurate to purposefully detrimental. To this end, we use toy functions and cheap surrogates, where priors of known quality can be obtained. Next, in Sec. 4.3, we compare πBO against two competitive approaches (BOPrO and BOWS) that integrate priors over the optimum similarly to πBO, using HyperMapper [33] as a parent framework, in which the most competitive baseline BOPrO is implemented. For these experiments we adopt a Multilayer Perceptron (MLP) benchmark on various datasets, using the interface provided by HPOBench [12], with priors constructed around the defaults provided by the library. Lastly, in Sec. 4.4, we apply πBO and other approaches to two deep learning tasks, also using priors derived from publicly available defaults.

4.1 Experimental Setup

Priors For our surrogate and toy function tasks, we follow the prior construction methodology in BOPrO [47] and create three main types of prior qualities, all Gaussian: strong, weak and wrong. The strong and weak priors are located to have a high and moderate density on the optimum, respectively. The wrong prior is a narrow distribution located in the worst region of the search space. For the OpenML MLP tuning benchmark, we utilize the defaults and search spaces provided in HPOBench [12], and construct Gaussian priors for each hyperparameter with their mean on the default value, and a standard deviation of 25% of the hyperparameter's domain. For the DL case studies, we utilize defaults from each task's repository and, for numerical hyperparameters, once again set the standard deviation to 25% of the hyperparameter's domain. For categorical hyperparameters, we place a higher probability on the default. As such, the quality of the prior is ultimately unknown, but serves as a proxy for what a practitioner may choose and has shown to be a reasonable choice [2]. For all experiments, we run πBO with $\beta = N/10$, where N is the total number of iterations, in order to make the prior influence approximately equal in all experiments, regardless of the number of allowed BO iterations.

Baselines We empirically evaluate πBO against the most competitive approaches for priors over the optimum described in Section 2.3: BOPrO [47] and BO in Warped Space (BOWS) [37]. To contextualize the performance of πBO, we provide additional, simpler baselines: random sampling, sampling from the prior and BO with prior-based initial design. The latter is initialized with the mode of the prior *in addition* to its regular initial design. In our main results, we choose Spearmint (with EI) [45] for this mode-initialized baseline, simply referring to it as Spearmint.

4.2 Robustness of πBO

First, we study the robustness of πBO. To this end, we show that πBO benefits from informative priors and can recover from wrong priors, being consistent with our theoretical results in Section 3.3. To this end, we consider a well-known black-box optimization function, Branin (2D), as well as two surrogate HPO tasks from the Profet suite [25]: FC-Net (6D) and XGBoost (8D). For these tasks, we exemplarily show results for πBO implemented in the Spearmint framework. As Figure 1.2 shows, πBO is able to quickly improve over sampling from the prior. Moreover, it improves substantially over Spearmint (with mode initialization) for all informative priors, staying up to an order of magnitude ahead throughout the optimization for both strong and weak priors. For wrong priors, πBO displays desired robustness by recovering to approximately equal regret as Spearmint. In contrast, Spearmint frequently fails to substantially improve from its initial design on the strong and weak prior, which demonstrates the importance of considering the prior throughout the optimization procedure. This effect is even more pronounced on the higher-dimensional tasks FCNet and XGBoost, where BO typically spends many iterations at the boundary [52]. Here, πBO rapidly improves multiple orders of magnitude over the initial design, displaying its ability to efficiently exploit the information provided by the prior.

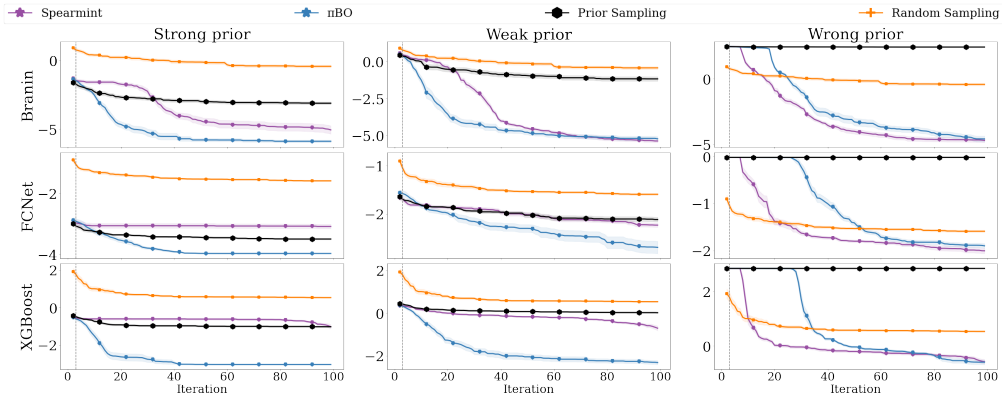

Fig. 1.2: Comparison of πBO, Spearmint, and two sampling approaches on Branin, FCNet and XGBoost for various prior strengths. Mean and standard error of log simple regret is displayed over 100 iterations, averaged over 20 runs. The vertical line represents the end of the initial design phase.

4.3 Comparison of πBO against other Prior-Guided Approaches

Next, we study the performance of πBO against other state-of-the-art prior-guided approaches. To this end, we consider optimizing 5 hyperparameters of an MLP for classification [12] on 6 different OpenML datasets [57] and compare against BOPrO [47] and BOWS [37]. For minimizing confounding factors, we implement πBO and BOWS in HyperMapper [33], the same framework that BOPrO runs on. Moreover, we let all approaches share πBO's initialization procedure. We consider a budget of 50 iterations as it is common with ML practitioners [6]. In Figure 1.3, we see that πBO offers the best performance on four out of six tasks, and displays the most consistent performance across tasks. In contrast to them BOWS and BOPrO, πBO also comes with theoretical guarantees and is flexible in the choice of framework and acquisition function.

4.4 Case Studies on Deep Learning Pipelines

Last, we study the impact of πBO on deep learning applications, which are often fairly expensive, making efficiency even more important than in HPO for traditional machine learning. To this end, we consider two deep learning case studies: segmentation of neuronal processes in electron microscopy images with a U-Net(6D) [39], with code provided from the NVIDIA deep learning examples repository [36], and image classification on ImageNette-128 (6D), a light-weight adaptation of ImageNet [11], with code from the repository of the popular FastAI

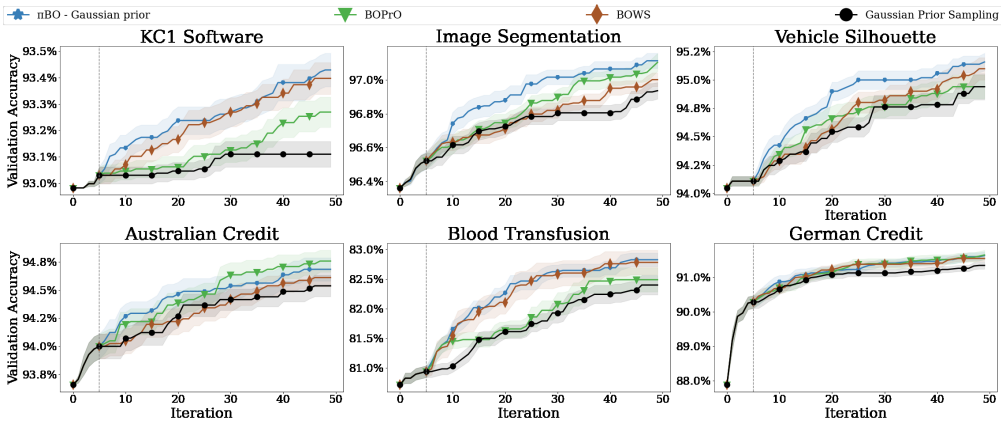

Fig. 1.3: Comparison of πBO, BOPrO, BOWS, and prior sampling for 5D MLP tuning on various OpenML datasets for a prior centered on default values. We show mean and standard error of the accuracy across 20 runs. The vertical line represents the end of the initial design phase.

library [19]. We mimic the setup from Section 4.3 by using the HyperMapper framework and identical initialization procedures across approaches. Gaussian priors are set on publicly available default values, which are results of previous tuning efforts of the original authors. We again optimize for a practical budget of 50 iterations [6]. As test splits for both tasks were not available to us, we report validation scores.

As shown in Figure 1.4, πBO achieves a 2.5× time-to-accuracy speedup over Vanilla BO. For ImageNette, the performance of πBO at iteration 4 already surpasses the performance of Vanilla BO at Iteration 50, demonstrating a 12.5× time-to-accuracy speedup. Ultimately, πBO's final performance establishes a new state-of-the-art validation performance on ImageNette with the provided pipeline, with a final accuracy of 94.14% (vs. the previous state of the art with 93.55%[1]).

[1] `https://github.com/fastai/imagenette#imagenette-leaderboard`, 80 Epochs, 128 Resolution

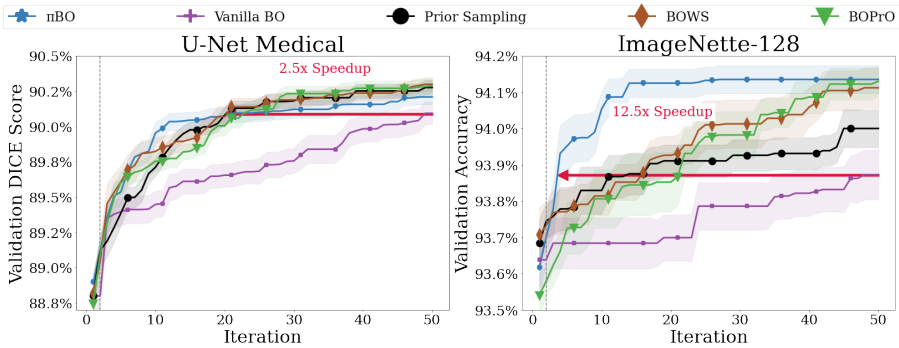

Fig. 1.4: Comparison of approaches for U-Net Medical and ImageNette-128 for a prior centered on default values. We show mean and standard error of the accuracy across 20 runs for U-Net Medical and 10 runs for ImageNette-128. The vertical line represents the end of the initial design phase.

5 Conclusion and Future Work

We presented πBO, a conceptually very simple Bayesian optimization approach for leveraging user beliefs about the location of an optimum, which relies on a generalization of myopic acquisition functions. πBO modifies the selection of design points through a decaying weighting scheme, promoting high-probability regions under the prior. Contrary to previous approaches, πBO imposes only minor restrictions on the type of priors, surrogates or frameworks that can be used. πBO provably converges at regular rates, displays state-of-the art performance across tasks, and effectively recovers from poorly specified priors. Moreover, we have demonstrated that πBO can yield substantial performance gains for practical low-budget settings, improving on the state-of-the-art for a real-world CNN tuning tasks even with trivial choices for the prior. For practitioners who have historically relied on manual or grid search for HPO, we hope that πBO will serve as an intuitive and effective tool for bridging the gap between traditional tuning methods and BO.

πBO sets the stage for several follow-up studies. Amongst others, we will examine the extension of πBO to non-myopic acquisition functions, such as entropy-based methods. Non-myopic acquisition functions do not fit well in the current πBO framework, as they do not necessarily benefit from evaluating inputs expected to perform well. We will also combine πBO with multi-fidelity optimization methods to yield even higher speedups, and with multi-objective optimization to jointly optimize performance and secondary objective functions, such as interpretability or fairness of models.

6 Ethics Statement

Our work proposes an acquisition function generalization which incorporates prior beliefs about the location of the optimum into optimization. The approach is foundational and thus will not bring direct societal or ethical consequences. However, πBO will likely be used in the development of applications for a wide range of areas and thus indirectly contribute to their impacts on society. In particular, we envision that πBO will impact a multitude of fields by allowing ML experts to inject their knowledge about the location of the optimum into Bayesian Optimization.

We also note that we intend for πBO to be a tool that allows users to assist Bayesian Optimization by providing reasonable prior knowledge and beliefs. This process induces user bias into the optimization, as πBO will inevitably start by optimizing around this prior. As some users may only be interested in optimizing in the direct neighborhood of their prior, πBO could allow them to do so if provided with a high β value in relation to the number of iterations. Thus, if improperly specified, πBO could serve to reinforce user's beliefs by providing improved solutions only for the user's region of interest. However, if used properly, πBO will reduce the computational resources required to find strong hyperparameter settings, contributing to the sustainability of machine learning.

7 Reproducibility

In order to make the experiments run in πBO as reproducible as possible, we have included links to repositories of our implementations in both Spearmint and HyperMapper, with instructions on how to run our experiments. Moreover, we have included in said repositories all of the exact priors that we have used for our runs, which run out of the box. The priors we used were, in our opinion, well motivated as to avoid subjectivity, which we hope serves as a good frame of reference for similar works in the future. Our Spearmint implementation of both πBO and BOWS is available at `https://github.com/piboauthors/PiBO-Spearmint`, and our HyperMapper implementation is available at `https://github.com/piboauthors/PiBO-Hypermapper`.

Acknowledgements

Luigi Nardi was supported in part by affiliate members and other supporters of the Stanford DAWN project — Ant Financial, Facebook, Google, Intel, Microsoft, NEC, SAP, Teradata, and VMware. Carl Hvarfner and Luigi Nardi were partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Artur Souza was supported by CAPES, CNPq, and FAPEMIG. Frank Hutter acknowledges support by the European Research Council (ERC) under the European Union Horizon 2020 research and innovation programme through grant no. 716721, through TAILOR, a project funded by the EU Horizon 2020 research and innovation programme under GA No 952215, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant number 417962828 and by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG. Marius Lindauer acknowledges support by the European Research Council (ERC) under the Europe Horizon programme. The computations were also enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at LUNARC partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

References

[1] K. Anand, Z. Wang, M. Loog, and J. van Gemert. Black magic in deep learning: How human skill impacts network training. In *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*. BMVA Press, 2020.

[2] M. Anastacio and H. Hoos. Combining sequential model-based algorithm configuration with default-guided probabilistic sampling. In *GECCO '20: Genetic and Evolutionary Computation Conference, Companion Volume, Cancún, Mexico, July 8-12, 2020*, pages 301–302. ACM, 2020.

[3] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. In *Advances in Neural Information Processing Systems*, 2020. URL `http://arxiv.org/abs/1910.06403`.

[4] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012.

[5] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Proceedings of the 25th International Conference on Advances in Neural Information Processing Systems (NeurIPS'11)*, pages 2546–2554, 2011.

[6] X. Bouthillier and G. Varoquaux. Survey of machine-learning experimental methods at NeurIPS2019 and ICLR2020. Research report, Inria Saclay Ile de France, 2020.

[7] E. Brochu, V. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv:1012.2599v1 [cs.LG]*, 2010.

[8] A. D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(88):2879–2904, 2011.

[9] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. Deisenroth. Bayesian gait optimization for bipedal locomotion. In P. Pardalos and M. Resende, editors, *Proceedings of the Eighth International Conference on Learning and Intelligent Optimization (LION'14)*, Lecture Notes in Computer Science. Springer, 2014.

[10] Y. Chen, A. Huang, Z. Wang, I. Antonoglou, J. Schrittwieser, D. Silver, and N. D. Freitas. Bayesian optimization in alphago. *ArXiv*, abs/1812.06855, 2018.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[12] Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, Rene Sass, Aaron Klein, Noor Awad, Marius Lindauer, and Frank Hutter. HPOBench: A collection of reproducible multi-fidelity benchmark problems for HPO. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL `https://openreview.net/forum?id=1k4rJYEwda-`.

[13] M. Feurer, Jost Tobias Springenberg, and F. Hutter. Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1128–1135, 2015.

[14] M. Feurer, B. Letham, F. Hutter, and E. Bakshy. Practical transfer learning for bayesian optimization. *ArXiv abs/1802.02219*, 2018.

[15] P. Frazier, W. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM J. Control and Optimization*, 47: 2410–2439, 01 2008. doi: 10.1137/070693424.

[16] The GPyOpt-authors. GPyOpt: A bayesian optimization framework in python. `http://github.com/SheffieldML/GPyOpt`, 2016.

[17] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(1):1809–1837, June 2012. ISSN 1532-4435.

[18] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, 2014. URL `https://proceedings.neurips.cc/paper/2014/file/069d3bb002acd8d7dd095917f9efe4cb-Paper.pdf`.

[19] J. Howard et al. fastai. `https://github.com/fastai/fastai`, 2018.

[20] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*, 2011.

[21] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 12 1998. doi: 10.1023/A:1008306431147.

[22] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

[23] K. Kandasamy, K. R. Vysyaraju, W. Neiswanger, B. Paria, C. R. Collins, J. Schneider, B. Poczos, and E. P. Xing. Tuning hyperparameters without grad students: Scalable and robust bayesian optimisation with dragonfly. *Journal of Machine Learning Research*, 21(81):1–27, 2020.

[24] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR*, 2015.

[25] A. Klein, Z. Dai, F. Hutter, N. Lawrence, and J. Gonzalez. Meta-surrogate benchmarking for hyperparameter optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[26] H. J. Kushner. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, 86(1):97–106, 03 1964. ISSN 0021-9223. doi: 10.1115/1.3653121. URL `https://doi.org/10.1115/1.3653121`.

[27] Eric Hans Lee, Valerio Perrone, Cedric Archambeau, and Matthias Seeger. Cost-aware bayesian optimization, 2020.

[28] C. Li, S. Rana, S. Gupta, V. Nguyen, S. Venkatesh, A. Sutti, D. R. de Celis, T. Slezak, M. Height, M. Mohammed, and I. Gibson. Accelerating experimental design by incorporating experimenter hunches. In *IEEE International Conference on Data Mining, ICDM*, pages 257–266. IEEE Computer Society, 2018.

[29] C. Li, S. Gupta, S. Rana, V. Nguyen, A. Robles-Kelly, and S. Venkatesh. Incorporating expert prior knowledge into experimental design via posterior sampling. *ArXiv*, abs/2002.11256, 2020.

[30] M. López-Ibáñez, J. Dubois-Lacoste, L. P. Cáceres, T. Stützle, and M. Birattari. The iRace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016.

[31] B. Matérn. Spatial variation. *Meddelanden fran Statens Skogsforskningsinstitut*, 1960.

[32] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

[33] L. Nardi, D. Koeplinger, and K. Olukotun. Practical design space exploration. In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 347–358. IEEE, 2019.

[34] C. Oh, E. Gavves, and M. Welling. BOCK : Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*, pages 3865–3874, 2018.

[35] V. Perrone, H. Shen, M. Seeger, C. Archambeau, and R. Jenatton. Learning search spaces for bayesian optimization: Another view of hyperparameter transfer learning. In *Advances in Neural Information Processing Systems*, 2019.

[36] S. Przemek et al. Nvidia deep learning examples. `https://github.com/NVIDIA/DeepLearningExamples`.

[37] A. Ramachandran, S. Gupta, S. Rana, C. Li, and S. Venkatesh. Incorporating expert prior in bayesian optimisation via space warping. *Knowledge-Based Systems*, 195:105663, 2020.

[38] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[39] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.

[40] Binxin Ru, Xingchen Wan, Xiaowen Dong, and Michael Osborne. Interpretable neural architecture search via bayesian optimisation with weisfeiler-lehman kernels. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=j9Rv7qdXjd`.

[41] B. Shahriari, A. Bouchard-Côté, and N. Freitas. Unbounded bayesian optimization via regularization. In *Artificial intelligence and statistics*, pages 1168–1176, 2016.

[42] B. Shahriari, K. Swersky, Z. Wang, R. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[43] E. Siivola, A. Vehtari, J. Vanhatalo, J. González, and M. R. Andersen. Correcting boundary over-exploration deficiencies in bayesian optimization with virtual derivative sign observations. In *International Workshop on Machine Learning for Signal Processing*, 2018.

[44] L. Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[45] J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian optimization of machine learning algorithms. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems (NeurIPS'12)*, pages 2960–2968, 2012.

[46] J. Snoek, K. Swersky, R. Zemel, and R. Adams. Input warping for bayesian optimization of non-stationary functions. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1674–1682. PMLR, 22–24 Jun 2014.

[47] A. Souza, L. Nardi, L. Oliveira, K. Olukotun, M. Lindauer, and F. Hutter. Bayesian optimization with a prior for the optimum. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part III*, volume 12977 of *Lecture Notes in Computer Science*, pages 265–296. Springer, 2021.

[48] J. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust Bayesian neural networks. In D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Proceedings of the 30th International Conference on Advances in Neural Information Processing Systems (NeurIPS'16)*, 2016.

[49] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, May 2012. ISSN 1557-9654. doi: 10.1109/tit.2011.2182033. URL `http://dx.doi.org/10.1109/TIT.2011.2182033`.

[50] D. Stoll, J. KH Franke, D. Wagner, S. Selg, and F. Hutter. Hyperparameter transfer across developer adjustments. In *NeurIPS 2020 Workshop on Meta-Learning*, 2020.

[51] K. Swersky, J. Snoek, and R. Adams. Multi-task Bayesian optimization. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Proceedings of the 27th International Conference on Advances in Neural Information Processing Systems (NeurIPS'13)*, pages 2004–2012, 2013.

[52] K. J. Swersky. *Improving Bayesian Optimization for Machine Learning using Expert Priors*. PhD thesis, University of Toronto, 2017.

[53] W. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[54] Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In Hugo Jair Escalante and Katja Hofmann, editors, *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, pages 3–26. PMLR, 06–12 Dec 2021. URL `https://proceedings.mlr.press/v133/turner21a.html`.

[55] Jan N. van Rijn and Frank Hutter. Hyperparameter importance across datasets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 2367–2376, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220058. URL `https://doi.org/10.1145/3219819.3220058`.

[56] J. Vanschoren. Meta-learning: A survey, 2018.

[57] J. Vanschoren, J. van Rijn, B. Bischl, and L. Torgo. OpenML: Networked science in machine learning. *SIGKDD Explor. Newsl.*, 15(2):49–60, 2014.

[58] Q. Wang, Y. Ming, Z. Jin, Q. Shen, D. Liu, M. J. Smith, K. Veeramachaneni, and H. Qu. Atmseer: Increasing transparency and controllability in automated machine learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12. Association for Computing Machinery, 2019.

[59] Z. Wang and S. Jegelka. Max-value entropy search for efficient bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 3627–3635. PMLR, 2017.

[60] M. Wistuba, N. Schilling, and L. Schmidt-Thieme. Hyperparameter search space pruning - A new component for sequential model-based hyperparameter optimization. In A. Appice, P. Rodrigues, V. Costa, J. Gama, A. Jorge, and C. Soares, editors, *Machine Learning and Knowledge Discovery in Databases (ECML/PKDD'15)*, volume 9285 of *Lecture Notes in Computer Science*, pages 104–119. Springer, 2015.

[61] L. Zimmer, M. Lindauer, and F. Hutter. Auto-pytorch tabular: Multi-fidelity metalearning for efficient and robust autodl. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):3079 – 3090, 2021.

Paper II

Joint Entropy Search for Maximally-Informed Bayesian Optimization

Carl Hvarfner
Lund University

Frank Hutter
University of Freiburg

Luigi Nardi
Lund University

Abstract

Bayesian optimization has become an established framework and popular tool for hyperparameter optimization (HPO) of machine learning (ML) algorithms. While known for its sample-efficiency, vanilla BO can not utilize readily available prior beliefs the practitioner has on the potential location of the optimum. Thus, BO disregards a valuable source of information, reducing its appeal to ML practitioners. To address this issue, we propose πBO, an acquisition function generalization which incorporates prior beliefs about the location of the optimum in the form of a probability distribution, provided by the user. In contrast to previous approaches, πBO is conceptually simple and can easily be integrated with existing libraries and many acquisition functions. We provide regret bounds when πBO is applied to the common Expected Improvement acquisition function and prove convergence at regular rates independently of the prior. Further, our experiments show that πBO outperforms competing approaches across a wide suite of benchmarks and prior characteristics. We also demonstrate that πBO improves on the state-of-the-art performance for a popular deep learning task, with a 12.5$\times$ time-to-accuracy speedup over prominent BO approaches.

1 Introduction

The optimization of expensive black-box functions is a prominent task, arising across a wide range of applications. *Bayesian optimization* (BO) [25, 35] is a sample-efficient approach, and has been successfully applied to various problems, including machine learning hyperparameter optimization [37, 2, 20, 33], robotics [6, 3, 23, 24], hardware design [27, 11], and tuning reinforcement learning agents like AlphaGo [7]. In BO, a probabilistic surrogate model is used for modeling the (unknown) objective. The selection policy employed by the BO algorithm is dictated by an acquisition function, which draws on the uncertainty of the surrogate to guide the selection of the next query. The choice of acquisition function is significant for the success of the BO algorithm.

A popular line of acquisition functions takes an information-theoretic angle, and considers the *expected information gain* regarding the location of the optimum that is obtained from an upcoming query. *Entropy Search* (ES) [15], *Predictive Entropy Search* (`PES`) [16] and the earlier work of IAGO [46] select queries by maximizing this quantity. While `ES` and `PES` are efficient in the number of queries to optimize the objective, they both require significant computational effort and complex approximations of the expected information gain, which impacts their performance and practical use [16, 47].

A related information-theoretic family of approaches considers the information gain on the optimal objective value [18, 47, 31]. *Max-value Entropy Search* (`MES`) [47] was the first information-theoretic approach to have a proven convergence rate, albeit only in a noiseless problem setting. Moreover, its consideration of a one-dimensional density over the output space as opposed to a D-dimensional input space and a reduction in intricate approximations yielded a computationally efficient alternative to the `ES`/`PES` line of approaches. Despite its empirical success, some crucial shortcomings of `MES` have been highlighted in recent works. Its convergence rate has been disputed [42], and crucially, it does not differentiate between the (unobserved) maximal objective value f^* and the observed noisy maximum y_{max} [41, 26, 28, 42]. As such, its assumption on the posterior distribution of the output $p(y|\mathcal{D}, \boldsymbol{x})$ does not hold in any setting where noise is present, and several follow-ups have been proposed to address the noisy problem setting [41, 26, 28, 42].

We propose an approach which merges the `ES`/`PES` and `MES` lines of work, and provides an all-encompassing perspective on information gain regarding optimality. We introduce Joint Entropy Search (JES), a novel acquisition function which has the following advantages over existing infomation-theoretic approaches:

1. It utilizes two sources of information, by considering the entropy over both the optimum and the noiseless optimal value;

2. It utilizes the full optimal observation, allowing it to rely primarily on exact computation through standard GP machinery instead of complex approximations; and

3. It is computationally light-weight, requiring minimal pre-computation relative to other information-theoretic approaches which consider the input space.

Simultaneously to our work, a similar approach aimed at the multi-objective setting, was proposed by [44]. The authors independently came up with the same JES acquisition function, with a subtly different approximation scheme to the one we present. We see our work as being complementary to theirs because we both demonstrate the effectiveness of this new acquisition function in different settings - theirs being multi-objective and batch evaluations, ours being single-objective and large levels of output noise. Our code for reproducing the experiments is available at`https://github.com/hvarfner/JointEntropySearch`.

2 Background and related work

Bayesian optimization. We consider the problem of optimizing a black-box function f across a set of feasible inputs $\mathcal{X} \subset \mathbb{R}^d$:

$$\boldsymbol{x}^* \in \arg\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}). \tag{2.1}$$

We assume that $f(\boldsymbol{x})$ is expensive to evaluate, and can potentially only be observed through a noise-corrupted estimate, y, where $y = f(\boldsymbol{x}) + \varepsilon, \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ for some noise level σ_ε^2. In this setting, we wish to maximize f in an efficient manner, typically while adhering to a budget which sets a cap on the number of points that can be evaluated. BO aims to globally maximize f by an initial design and thereafter sequentially choosing new points $\boldsymbol{x}_n$ for some iteration n, creating the data $\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{(\boldsymbol{x}_n, y_n)\}$. After each new observation, BO constructs a probabilistic surrogate model $p(f|\mathcal{D}_n)$ and uses that surrogate to build an acquisition function $\alpha(\boldsymbol{x}, \mathcal{D}_n)$. The combination of surrogate model and acquisition function encodes the strategy for selecting the next point $\boldsymbol{x}_{n+1}$. After the full budget of N iterations is exhausted, a best configuration $\boldsymbol{x}_N^*$ is returned as either the arg max of the observed values, or the optimum as predicted by the surrogate model.

Gaussian processes. When constructing the surrogate, the most common choice is *Gaussian processes* (GPs) [30]. Formally, a GP is an infinite collection of random variables, such that every finite subset of those variables follows a multivariate Gaussian distribution. The GP utilizes a covariance function k, which encodes a prior belief for the smoothness of f, and determines how previous observations influence prediction. Given observations $\mathcal{D}_n$ at iteration n, the posterior $p(f|\mathcal{D}_n)$ over the objective is characterized by the posterior mean m_n and variance s_n of the GP:

$$m_n(\boldsymbol{x}) = \mathbf{k}_n(\boldsymbol{x})^\top (\mathbf{K}_n + \sigma_\varepsilon^2 \mathbf{I})^{-1} \mathbf{y}, \tag{2.2}$$

$$s_n(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \mathbf{k}_n(\boldsymbol{x})^\top (\mathbf{K}_n + \sigma_\varepsilon^2 \mathbf{I})^{-1} \mathbf{k}_n(\boldsymbol{x}), \tag{2.3}$$

where $(\mathbf{K}_n)_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\mathbf{k}_n(\boldsymbol{x}) = [k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_n)]^\top$ and σ_ε^2 is the noise variance. Alternative surrogate models include random forests [19] and Bayesian neural networks [38, 39].

Acquisition functions. The *acquisition function* acts on the surrogate model to quantify the attractiveness of a point in the search space. Acquisition functions employ a trade-off between exploration and exploitation, typically using a greedy heuristic to do so. Simple, computationally cheap heuristics are Expected Improvement (`EI`) [21, 5]. For a noiseless function, `EI` selects the next point $\boldsymbol{x}_{n+1}$ as

$$\boldsymbol{x}_{n+1} \in \arg\max_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}\left[(y_n^* - y_{n+1}^*)^+\right] = \arg\max_{\boldsymbol{x} \in \mathcal{X}} Z s_n(\boldsymbol{x}) \Phi(Z) + s_n(\boldsymbol{x}) \phi(Z), \tag{2.4}$$

where $Z = (y_n^* - m_n(\boldsymbol{x}))/s_n(\boldsymbol{x})$. Other acquisition functions which use similar heuristics are the Upper Confidence Bound (`UCB`) [40], and Probability of Improvement (PI) [22]. A more sophisticated approach related to `EI` is Knowledge Gradient (KG) [12].

Information-theoretic acquisition functions. Information-theoretic acquisition functions [15, 16, 32, 47] and their various adaptations [34, 17, 1] seek to maximize the expected information gain I from observing a subsequent query $(\boldsymbol{x}, y)$ regarding the optimum, $\boldsymbol{x}^*$. This equates to reducing the uncertainty of the density over the optimum, $p(\boldsymbol{x}^*|\mathcal{D}) = \mathbb{P}(\boldsymbol{x} = \arg\max_{\boldsymbol{x}' \in \mathcal{X}} f(\boldsymbol{x}')|\mathcal{D})$, using the information obtained through $(\boldsymbol{x}, y)$. By quantifying uncertainty through the differential entropy H, design points are selected based on the expected reduction in this quantity over $p(\boldsymbol{x}^*|\mathcal{D})$. Formally, this is expressed as the difference

between the current entropy over $p(\boldsymbol{x}^*|\mathcal{D})$, and the expected entropy of that density after observing the next query:

$$\alpha_{\texttt{ES}}(\boldsymbol{x}) = I((\boldsymbol{x}, y); \boldsymbol{x}^*|\mathcal{D}) = \mathrm{H}[p(\boldsymbol{x}^*|\mathcal{D})] - \mathbb{E}_y \left[\mathrm{H}[p(\boldsymbol{x}^*|\mathcal{D} \cup (\boldsymbol{x}, y)]\right]. \tag{2.5}$$

By utilizing the symmetric property of the mutual information, one can arrive at an equivalent expression, where the entropy is computed with regard to the density over the output y,

$$\alpha_{\texttt{PES}}(\boldsymbol{x}) = I(y; (\boldsymbol{x}, \boldsymbol{x}^*)|\mathcal{D}) = \mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}^*} \left[\mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x}, \boldsymbol{x}^*)]\right]. \tag{2.6}$$

Eq. 2.5 is the original formulation used in ES [15] and Eq. 2.6 is the formulation introduced with PES [16]. Both formulations require a series of approximations and expensive computational steps to compute the entropy in the second term. For PES specifically, with n data points of dimension d, the second term is estimated through Monte Carlo (MC) methods by computing Cholesky decompositions of size $\mathcal{O}(n + d^2/2)^3$, and approximating the Hessian at the optimum for each MC sample.

MES [47] avoids this computational hurdle by considering the information gain $I((\boldsymbol{x}, y); y^*|\mathcal{D})$ regarding the optimal value y^*. As such, it computes the entropy reduction for a one-dimensional density:

$$\alpha_{\texttt{MES}}(\boldsymbol{x}) = I(y; (\boldsymbol{x}, y^*)|\mathcal{D}) = \mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x})] - \mathbb{E}_{y^*} \left[\mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x}, y^*)]\right]. \tag{2.7}$$

Here, it is assumed that the posterior predictive distribution $p(y|\mathcal{D}, \boldsymbol{x}, y^*)$ is a truncated Gaussian distribution, for which the entropy can be computed in closed form. However, $p(y|\mathcal{D}, \boldsymbol{x}, y^*)$ takes this form only in a strictly noiseless setting [41, 28], where it holds true that $f^* = y_{max}$, i.e. when the maximal observation and the optimal value of the objective function coincide. For noisy applications, this assumption leads to an overestimation of the entropy reduction [28].

3 Joint Entropy Search

We now present Joint Entropy Search (JES), a novel information-theoretic approach for Bayesian optimization. As for other information-theoretic acquisition functions, JES considers a mutual information quantity. However, unlike its predecessors, JES adds an additional piece of information: compared to ES/PES, it adds the density over the noiseless optimal value f^*, and compared to MES, it adds the density over $\boldsymbol{x}^*$. It utilizes a novel two-step reduction in the predictive entropy from conditioning on sampled optima and their associated values. Throughout the section, we will refer to a sampled optimum and its associated value, $(\boldsymbol{x}^*, f^*)$, as an *optimal pair*.

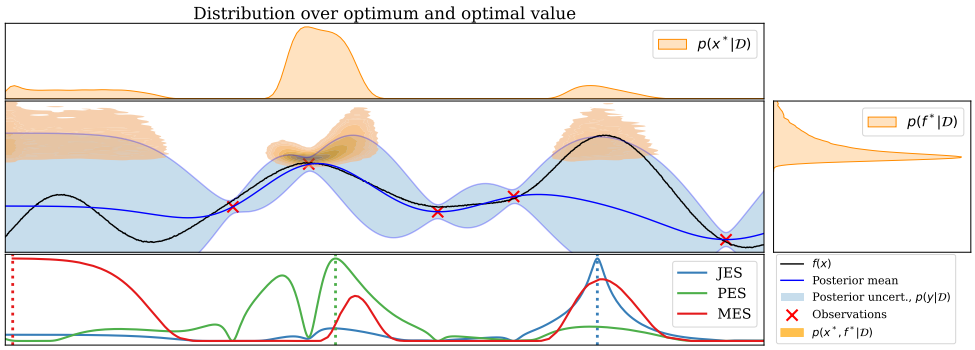

Fig. 2.1: The densities considered by ES/PES (top), MES (right) and JES (center) on a one-dimensional toy example. The multimodal density $p(\boldsymbol{x}^*, f^*)$ is reduced to a heavy-tailed density over f^* for the density used by MES (right), which does not capture the multi-modality of the density over the optimum. The density over $\boldsymbol{x}^*$ used by PES (top) does not capture the apparent exploration/exploitation trade-off that exists between the modes. The acquisition functions and their next point selections are highlighted with dashed lines (bottom).

3.1 Joint density over the optimum and optimal value

JES considers the joint probability density $p(\boldsymbol{x}^*, f^*)$ over both the optimum $\boldsymbol{x}^*$ and the true, noiseless optimal value f^*. Fig. 2.1 visualizes the densities $p(\boldsymbol{x}^*)$ and $p(f^*)$, considered by ES/PES and MES, respectively, and the joint density $p(\boldsymbol{x}^*, f^*)$, considered by JES. As highlighted by the vertical dashed lines for the point selection of each strategy (bottom), PES chooses strictly to reduce the uncertainty over $\boldsymbol{x}^*$, and as such, considers a region where the uncertainty over the optimal value is low. However, it can effectively determine that the right side of the local optimum is more promising to query next. MES seeks to reduce the tail of the probability density over f^* (right), which in this case leads to an exploratory query. JES' joint probability density over optimum and optimal value captures uncertainties over both "where" and "how large" the optimum will be. As such, it selects a point which is uncertain under both measures. As such, JES will learn about likely locations for the optimum, while simultaneously learning probable lower and upper bounds for the optimal value, which by itself yields an effective query strategy [47] and provides valuable knowledge for future queries. For the selected query in Fig. 2.1, JES will learn substantially about both $\boldsymbol{x}^*$ and f^* by querying it, whereas PES and MES learn only about one of them.

3.2 The Joint Entropy Search acquisition function

We consider the mutual information between the random variables $(\boldsymbol{x}^*, f^*)$ and a future query $(\boldsymbol{x}, y)$:

$$\alpha_{\texttt{JES}}(\boldsymbol{x}) = I((\boldsymbol{x}, y); (\boldsymbol{x}^*, f^*)|\mathcal{D}_n) \tag{2.8}$$

$$= \mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x})] - \mathbb{E}_{(\boldsymbol{x}^*, f^*)}\left[\mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x}, \boldsymbol{x}^*, f^*)]\right] \tag{2.9}$$

$$= \mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x})] - \mathbb{E}_{(\boldsymbol{x}^*, f^*)}\left[\mathrm{H}[p(y|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*)]\right]. \tag{2.10}$$

Eq. 2.9 is similar to Eq. 2.7 but with the addition of $\boldsymbol{x}^*$ and the replacement of y^* with f^* in the conditioning of the second term. The expectation is computed with respect to a $D+1$-dimensional joint probability density over $\boldsymbol{x}^*$ and f^*. In Eq. 2.10, we make it explicit that the conditional density inside the expectation is obtained after 1. conditioning the GP on the previous data $\mathcal{D}$, plus one additional noiseless optimal pair $(\boldsymbol{x}^*, f^*)$, and 2. knowing that the noiseless optimal value is in fact f^*. By utilizing the complete observation $(\boldsymbol{x}^*, f^*)$, we can treat it like any (noiseless) observation. As such, we quantify much of the entropy reduction by utilizing standard GP conditioning functionality. For 2., we cannot globally condition on $f(\boldsymbol{x}') \leqslant f^*, \forall \boldsymbol{x}'$. As such, we follow previous work [28, 47, 26, 42] and enforce the condition *locally* at the current query $\boldsymbol{x}$. The resulting effect is to truncate the GP's posterior over f locally at $\boldsymbol{x}$, upper bounding it to f^*. Notably, utilizing the fantasized observation $(\boldsymbol{x}^*, f^*)$ guarantees that the conditioned optimal value f^* in `JES` is actually obtained, rather than serving as a possibly unattained upper bound, which is typical in the MES family of acquisition functions. The expectation in Eq. 2.10 is approximated through MC by sampling L optimal pairs $\{(\boldsymbol{x}^*_\ell, f^*_\ell)\}_{\ell=1}^L$ from $p(\boldsymbol{x}^*, f^*)$ using an approximate version of *Thompson Sampling* (`TS`) [43], as explained in Sec. 3.3. In Fig. 2.2, the resulting posterior distribution of the two-step conditioning is shown in greater detail. As pointed out in [41, 28], after conditioning on f^*, the posterior predictive density over y is a sum of a truncated Gaussian distribution over f and the Gaussian noise ε. The entropy reduction from the two-step conditioning yields two separate variance reduction steps over $p(y|\mathcal{D}, \boldsymbol{x})$: a conditioning term and a truncation term. The former is computed exactly, while the latter, generally smaller term, requires approximation, as shown in Sec. 3.4.

Fig. 2.3 shows the difference in log variance over $p(y|\mathcal{D}, \boldsymbol{x})$ resulting from conditioning (in blue) and truncation (in orange) for the scenario in Fig. 2.2. The overall reduction is largest close to the point of conditioning, and the truncation term mainly contributes at uncertain regions far away from the conditioned point. Moreover, the magnitude of the conditioning term will rely on the prior

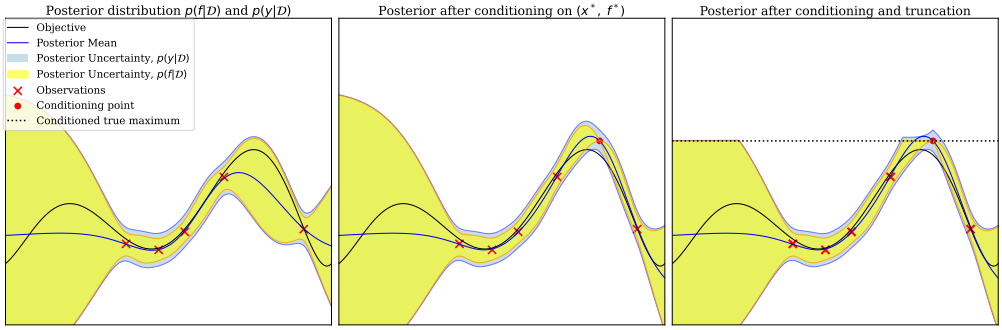

Fig. 2.2: Step-by-step modeling when conditioning on one optimal pair $(\boldsymbol{x}^*, f^*)$. The posterior with noise $p(y|\mathcal{D})$ and without noise $p(f|\mathcal{D})$ are illustrated in blue and yellow, respectively. The GP after 5 (noisy) observations, before conditioning on $(\boldsymbol{x}^*, f^*)$ is shown on the left. In the middle panel, we draw $(\boldsymbol{x}^*, f^*)$ and condition on it, making $p(f|\mathcal{D} \cup (\boldsymbol{x}^*, f^*))$ a delta distribution at the conditioning point as the fantasized observation f^* is noiseless. Since f^* is also the presumed noiseless maximum, we truncate its posterior $p(f|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), f^*)$ globally in the right panel. The observation noise allows for non-zero density on $p(y > f^*|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), f^*)$. We note that, while the noise is homoscedastic, its relative contribution to the total variance differs over the input space. As such, and since we're plotting standard deviations (not variances), the blue region is wider near observed data, where $p(f|\mathcal{D})$ has lower variance.

variance at the conditioned point, as a larger prior variance will lead to a larger reduction in entropy from conditioning. As we average over optimal pairs, many such entropy reduction terms accumulate.

3.3 Incorporating optimal pairs

To obtain samples $(\boldsymbol{x}^*, f^*)$, we utilize an approximate variant of `TS` [43], originally proposed in `PES` [16]. We utilize Bochner's theorem [4], which, for any stationary kernel k, asserts the existence of its Fourier dual $s(\boldsymbol{w})$. By normalizing $s(\boldsymbol{w})$, we obtain the spectral density $p(\boldsymbol{w}) = s(\boldsymbol{w})/\alpha$, where α is a normalization constant. We can then write the kernel as an expectation,

$$k(\boldsymbol{x}, \boldsymbol{x}') = \alpha \mathbb{E}_{\boldsymbol{w}}[e^{i\boldsymbol{w}^\intercal(\boldsymbol{x}-\boldsymbol{x}')}] = 2\alpha \mathbb{E}_{\boldsymbol{w},\boldsymbol{b}}[\cos(\boldsymbol{w}^\intercal \boldsymbol{x} + \boldsymbol{b}) \cos(\boldsymbol{w}^\intercal \boldsymbol{x}' + \boldsymbol{b})], \quad (2.11)$$

where $\boldsymbol{b} \sim \mathcal{U}(\boldsymbol{0}, 2\pi \boldsymbol{I})$. Following [29], we sample $\boldsymbol{b}$ and $\boldsymbol{w}$ to obtain an unbiased estimate of the kernel k. From this approximation, approximate sample paths can be drawn as a weighted sum of basis functions. This form allows for fast and dense querying of the sample paths – the arg max and max of which is an approximate draw from $p(\boldsymbol{x}^*, f^*)$. In `PES`, each sample $\boldsymbol{x}_\ell^*$ along with its inverted Hessian is required for computing the acquisition function. To obtain the

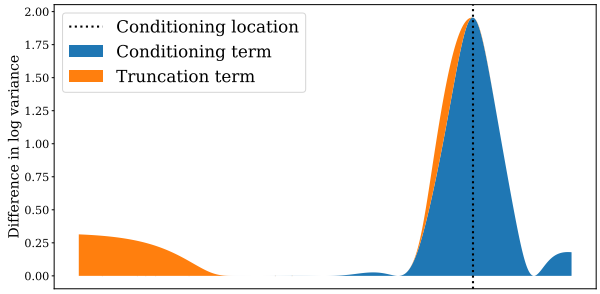

Fig. 2.3: Reduction in log variance from the conditioning step and the truncation step as visualized in Fig. 2.2. The local conditioning term (blue), and the globally variance-reducing truncation term (orange).

Hessian, each sample needs to be thoroughly optimized through gradient-based optimization. JES on the other hand, only requires $(\boldsymbol{x}^*, f^*)$. As such, it can rely on cheap, approximate optimization of these samples, e.g., by densely querying sample points on a non-uniform grid.

After obtaining a set of optimal pairs $\{(\boldsymbol{x}_\ell^*, y_\ell^*)\}_{\ell=1}^L$, JES computes the conditional entropy quantity over the output y. Concretely, we generate L GPs, each modeling a posterior density $\{p(y|\mathcal{D} \cup (\boldsymbol{x}_\ell^*, f_\ell^*), \boldsymbol{x})\}_{\ell=1}^L$ conditioned on an optimal pair and previously observed data $\mathcal{D}$. Since each optimal pair is drawn from the current GP hyperparameter set, we know that the current hyperparameter set is the correct one even after adding the optimal pair to the data. By consequence, JES can compute the updated inverse Gram matrix, $(K + \sigma_\varepsilon^2 I)^{-1}$, through a rank-1 update, instead of solving a linear system of equations. Utilizing the Sherman–Morrison formula [36], we obtain updated Gram matrices in $\mathcal{O}(n^2)$ for each sample, as opposed to $\mathcal{O}(n^3)$ for solving the linear system of equations.

3.4 Approximating the truncated entropy

As highlighted in the right panel of Fig. 2.2, conditioning on f^* yields a truncated normal distribution $p(f|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*)$ after having locally enforced the inequality $f(\boldsymbol{x}) \leqslant f^*$. The entropy, however, is computed with regard to the density over noisy observations, $y = f + \varepsilon$, which follows an Extended Skew distribution [28] and as such, does not have tractable entropy. We approximate this quantity through moment matching [26] of the truncated Gaussian distribution over f, which yields a valid lower bound on the information gain [26]. Consequently, we obtain two Gaussian densities $\hat{p}(f|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*) \sim \mathcal{N}(m_{f|f^*}, \sigma^2_{f|f^*})$ and $p(\varepsilon) \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, where

$m_{f|f*}$ and $\sigma^2_{f|f*}$ are the mean and variance of the truncated Gaussian posterior $p(f|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*)$. Due to independence between f and ε and the linearity of Gaussian distributions, we can then compute the entropy of the approximate density $\hat{p}_y$ exactly as $\mathrm{H}[\hat{p}(y|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*)] = \log(2\pi(\sigma^2_\varepsilon + \sigma^2_{f|f*}))$. Moreover, the variance of the truncated Gaussian $\sigma^2_{f|f*}$ is computed as

$$\sigma^2_{f|f*}(\boldsymbol{x}; \mathcal{D} \cup (\boldsymbol{x}^*_\ell, f^*_\ell)) = \sigma^2_T(f^*; m^\ell_n(\boldsymbol{x}), s^\ell_n(\boldsymbol{x})) \tag{2.12}$$

where $\sigma^2_T(\alpha; \mu, \sigma^2)$ is the variance of an upper truncated Gaussian distribution with parameters (μ, σ^2), truncated at α, and $m^\ell_n(\boldsymbol{x})$ and $s^\ell_n(\boldsymbol{x})$ are the mean and covariance functions of the GP which is conditioned on the optimal pair $(\boldsymbol{x}^*_\ell, f^*_\ell)$.

3.5 Exploitative selection to guard against model misspecification

As with all information-theoretic approaches, `JES` aims to reduce the uncertainty over the location of the optimum. With this strategy, the incentive to query the perceived optimum is often lower than for heuristic approaches, such as `EI`. In cases where the surrogate model is misspecified, information-theoretic approaches risk reducing the entropy based on a faulty belief of the optimum, which can drastically impact their performance. As a remedy, we utilize a γ-exploit approach inspired by the parallel context of AEGIS [9]: with probability γ, `JES` will query the arg max of the posterior mean to confirm its belief of the location of the optimum. If the model is misspecified, these exploitative steps enable the algorithm to reconsider its beliefs, rather than continuing to act based on faulty ones. This approach can substantially improve performance in cases of surrogate model misspecification, while having negligible impact on performance in the worst case.

3.6 Putting it all together: The `JES` algorithm

For a sampled set of size L, containing optimal pairs $\{(\boldsymbol{x}^*_\ell, y^*_\ell)\}_{\ell=1}^L$ and GPs with mean and covariance functions $\{m^\ell_n(\boldsymbol{x}), s^\ell_n(\boldsymbol{x})\}_{\ell=1}^L$, the expression for the `JES` acquisition function is

$$\alpha(\boldsymbol{x})_{\texttt{JES}} = \mathrm{H}[p(y|\mathcal{D}, \boldsymbol{x})] - \mathbb{E}_{(\boldsymbol{x}^*, f^*)}\left[\mathrm{H}[p(y|\mathcal{D} \cup (\boldsymbol{x}^*, f^*), \boldsymbol{x}, f^*)]\right] \tag{2.13}$$

$$\approx \log(2\pi(s_n(\boldsymbol{x}) + \sigma^2_\varepsilon)) \tag{2.14}$$

$$- \frac{1}{L}\sum_{\ell=1}^{L} \log(2\pi(\sigma^2_\varepsilon + \sigma^2_{f|f*}(\boldsymbol{x}; \mathcal{D} \cup (\boldsymbol{x}^*_\ell, f^*_\ell)))), \tag{2.15}$$

Algorithm 2.1 `JES` Algorithm

Input: Black-box function f, input space $\mathcal{X}$, size M of the initial design, max number of optimization iterations N, number of posterior MC samples L, fraction of exploit samples γ.
Output: Optimized design $\boldsymbol{x}^*$.
$\mathcal{D}_M \leftarrow \{(\boldsymbol{x}_i, y_i)\}_{i=1}^M$ `{initial design}`
for $\{n = M+1, \ldots, M+N\}$ **do**
  $m(\boldsymbol{x}), s^2(\boldsymbol{x}) \leftarrow \mathrm{FitGP}(\mathcal{D}_{n-1})$
  if $\mathrm{Rand}(0,1) \leqslant \gamma$ **then**
    $\boldsymbol{x}_n \leftarrow \arg\max_{\boldsymbol{x}\in\mathcal{X}} m_n(\boldsymbol{x})$ `{as described in Sec. 3.5}`
  else
    for $\{\ell = 1, \ldots, L\}$ **do**
      $(\boldsymbol{x}_\ell^*, y_\ell^*) \leftarrow \mathtt{TS}(f)$ `{as described in Sec. 3.3}`
      $p(y|\mathcal{D}_{n-1} \cup (\boldsymbol{x}_\ell^*, f_\ell^*), \boldsymbol{x}) \leftarrow UpdateGP(\boldsymbol{x}_\ell^*, f_\ell^*)$ `{as described in Sec. 3.3}`
    end for
    $\boldsymbol{x}_n = \arg\max_{\mathcal{X}} \alpha_{\mathtt{JES}}(\boldsymbol{x})$ `{defined in Eq. 2.13}`
  end if
  $y_n = f(\boldsymbol{x}_n) + \varepsilon, \quad \mathcal{D}_n \leftarrow \mathcal{D}_{n-1} \cup \{(\boldsymbol{x}_n, y_n)\}$ `{observe next query}`
end for
return $\boldsymbol{x}^* \leftarrow \arg\max_{\boldsymbol{x}\in\mathcal{X}} m_n(\boldsymbol{x})$

The first term in 2.15 is simply the entropy of a Gaussian that can be computed in closed form. The second term contains both the conditioning term, which is exact, and the truncation, which is approximated as described in Sec. 3.4. Algorithm 2.1 outlines pseudocode for `JES` in its entirety.

4 Experimental evaluation

Benchmarks. We now evaluate `JES` on a suite of diverse tasks. We consider three different types of benchmarks: samples drawn from a GP prior, commonly used synthetic test functions [16], and a collection of classification tasks on tabular data using an MLP, provided through HPOBench [10]. For the GP prior tasks, the hyperparameters are known for all methods to evaluate the effect of the acquisition function in isolation. Consequently, we do not use the γ-exploit approach from Sec. 3.5 in this case (i.e., we set $\gamma = 0$ in Algorithm 2.1). For the synthetic and MLP tasks, we marginalize over the GP hyperparameters, and set $\gamma = 0.1$.

Evaluation criteria. We use two types of evaluation criteria as in [47]: *simple regret* and *inference regret*. The simple regret $r_n = \max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) - \max_{t \in [1,n]} f(\boldsymbol{x}_t)$ measures the value of the best queried point so far. After a query, we may infer an arg max of the function, which is chosen as $\boldsymbol{x}_n^* = \arg\max_{\boldsymbol{x} \in \mathcal{X}} m_n(\boldsymbol{x})$ [15, 47, 16]. We denote the inference regret as $r_n = \max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) - f(\boldsymbol{x}_n^*)$. Since information-theoretic approaches do not necessarily seek to query the optimum, but only to know its location, inference regret characterizes how satisfying our belief of the arg max is. Notably, this metric is non-monotonic, meaning that the best guess can worsen with time. We use this metric in the ideal model benchmarking setting, when we sample tasks from a GP with known hyperparameters. We use simple regret for the synthetic test functions, as it constitutes a metric that is more robust to surrogate model misspecification.

The experimental setup. We compare against other state-of-the-art information-theoretic approaches: `PES` [16] and `MES` [47], as well as `EI` [21]. The acquisition functions are all run in the same framework written in MATLAB, created for the original `PES` implementation by [16]. All synthetic experiments were run for $50D$ iterations. In the main paper, we fix the number of MC samples for MES, PES and JES to 100 each.

4.1 GP prior samples

We consider samples from a GP prior for four different dimensionalities: 2D, 4D, 6D, and 12D, with a noise standard deviation of 0.1 for a range of outputs spanning roughly $[-10, 10]$. These tasks constitute an optimal setting for each algorithm, as the surrogate perfectly models the task at hand. In Fig. 2.4, `JES` demonstrates empirically the value of the additional source of information, substantially outperforming `PES` and `MES` on all tasks.

Fig. 2.5 compares `JES` (top left) against `PES`, `MES` and `EI` in terms of point selection for one repetition on a two-dimensional sample task, where all runs are initialized with $D + 1$ identical random samples. We observe that `JES` succeeds in finding all attractive regions of the search space, and queries the region around the optimum densely, which is sensible in a noisy setting. We further notice that `EI` (bottom right) fails to query the two circled local optima. `PES` (bottom left) also ignores two local optima to various degrees, and tends to circle the (perceived) optimum densely, which is expensive in terms of number of evaluations. We believe this showcases a shortcoming of only considering the density over the optimum: `PES` circles the optimum, but does not query its value. Lastly, `MES` (top right) successfully queries all attractive regions of the space,

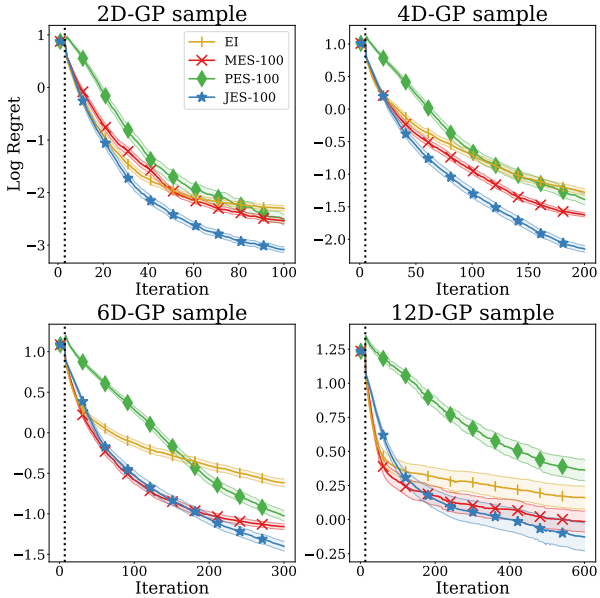

Fig. 2.4: Comparison of `JES`, `MES`, `PES` and `EI` on GP prior samples. We run 1000 repetitions each for 2, 4 and 6D, and 250 on 12D. Mean and 2 standard errors of log regret are displayed for each acquisition function. The vertical dashed line shows the end of the initial design phase.

but also samples regions that are evidently poor the most densely out of the four approaches, despite information given by earlier (brighter) samples. Since `JES` considers the information conveyed by both `MES` and `PES`, it successfully excludes the apparent suboptimal regions of the space, finds all relevant optima, and queries these optima in a desirable manner.

We additionally evaluate the performance of all approaches on GP sample tasks that have a substantial amount of noise - its standard deviation roughly accounting for 10% of the total output range. We run these tasks with the GP hyperparameters fixed a priori for a larger number of iterations, 125D, to display the stagnation of some approaches. While `MES` and `PES` slow down approximately at the halfway point for both tasks, `JES` steadily improves for the entire length of the run. This robustness to large noise magnitudes highlights the importance of intrinsically handling noisy objectives in `JES`.

In Table 2.2, we display the runtime of each acquisition function on these tasks when marginalizing over 10 sets hyperparameters, and sampling 10 optima per set. We time each iteration from after hyperparameters have been sampled, up until (but excluding) the query of the black-box function. Thus, acquisition

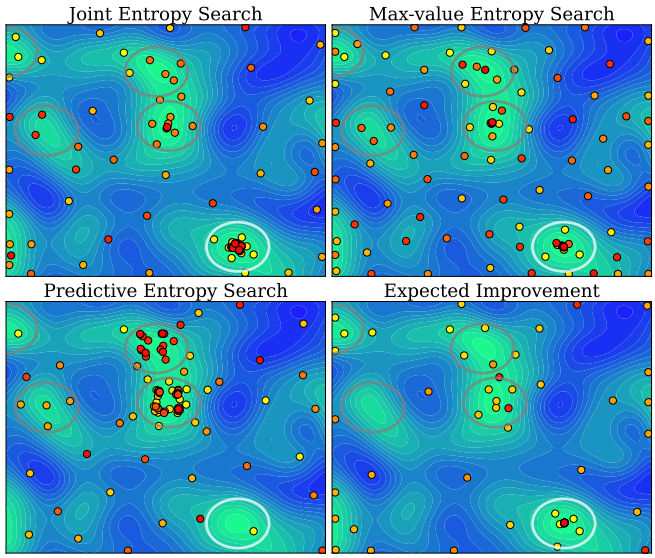

Fig. 2.5: Comparison of queries for `JES` (top left), `MES` (top right), `PES` (bottom left) and `EI` (bottom right) on a sample of a 2D GP after a 100 function evaluations. The global optimum is circled in white, and four local optima in gray. Earlier queries are colored yellow, and later queries red.

function pre-computation and optimization are included. The runtime of `JES` is only marginally slower than that of `MES` with Gumbel sampling, while being at least an order of magnitude faster than `PES` for all displayed dimensionalities.

4.2 Synthetic test functions

Next, in Fig. 2.7, we study the performance of `JES` on three optimization test functions: Branin (2D), Hartmann (3D) and Hartmann (6D). For these tasks, we follow convention [16, 31] and marginalize over GP hyperparameters. On Branin, `JES` starts out slightly slower than MES but reaches the same performance in 100 iterations; and on the two Hartmann functions, JES performs amongst the best in the beginning and clearly best in the end. We note that `PES` experienced numerical issues on Branin, and as such, we acknowledge that its performance should be better than what is reported.

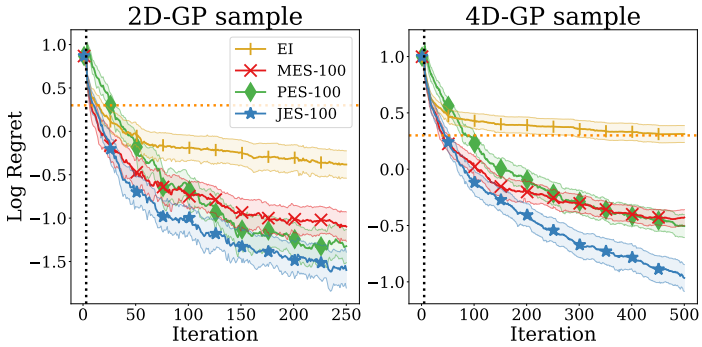

Fig. 2.6: Evaluation of `JES`, `MES`, `PES` and `EI` on noisy ($\sigma_\varepsilon^2 = 4$, orange) GP sample tasks across 100 repetitions. Mean and 2 standard errors of log regret are displayed for each acquisition function.

4.3 MLP tasks

Lastly, we evaluate the performance of `JES` on the tuning an MLP model's 4 hyperparameters for 20D iterations on six datasets. These tasks are part of the OpenML[2] library of tasks, and the HPO benchmark is provided through the HPOBench [10] suite. We measure the best observed classification accuracy. Notably, these tasks have a large amount of noise, which causes the performance to fluctuate substantially between repetitions. We observe that `JES` performs substantially better on two tasks, and is approximately equal in performance to `EI` on three, with `EI` being superior in one task. JES displays superior or equal performance to `MES` on all tasks, with `PES` lagging behind.

5 Conclusions

We have presented Joint Entropy Search, an information-theoretic acquisition function that considers an entirely new quantity, namely the joint density over the optimum and optimal value. By utilizing the entropy reduction from fantasized optimal observations, `JES` obtains a simple form for the entropy reduction regarding the joint distribution. As such, the additional information considered comes with minimal computational overhead, avoids restrictive assumptions on the objective, and yields state-of-the-art performance along with superior decision-making. We believe `JES` to be a new go-to acquisition function for BO, and to establish a new standard for subsequent information-theoretic techniques.

[2]https://www.openml.org/

Tab. 2.2: Runtime of `JES`, `MES`, `PES` and `EI` on GP sample tasks of varying dimensionalities. JES is only marginally slower than MES, and orders of magnitude faster than PES.

Task	JES-100	MES-100	PES-100	EI
2D	1.40 ± 0.32	1.03 ± 0.19	17.39 ± 4.95	0.23 ± 0.13
4D	1.50 ± 0.37	1.21 ± 0.3	34.53 ± 8.3	0.3 ± 0.17
6D	1.56 ± 0.39	1.26 ± 0.37	62.92 ± 13.54	0.35 ± 0.2

6 Limitations and Future Work

The main contribution of this paper is to provide a novel information-theoretic acquisition function which, given a sufficiently accurate model, yields impressive results. However, the non-myopic, speculative nature of information-theoretic approaches lend them to be susceptible to model misspecification, such as a poor choice of GP kernel or GP hyperparameters. In our view, information-theoretic approaches are possibly more susceptible to this issue than their myopic counterparts (`EI`, `UCB`, `TS`). While we propose a remedy to stabilize and improve the acquisition function under model misspecification with the γ-exploit approach, this technique only serves to *discover* misspecification and adjust accordingly, but not to inherently fix the misspecification. We believe misspecification can only be remedied by altering the surrogate model. It is thus very promising to combine advanced modelling techniques with information-theoretic acquisition functions, as already done with the additive GP approach utilized in conjunction with `MES` by [47]; further promising additions would be to tackle heterogeneous noise and input warping as done by HEBO [8].

We also note that, since `JES` computes the entropy reduction from conditioning on the optimal pair, it relies on some level of noise in the objective. A surrogate model with zero noise will result in an infinite information gain for every optimal pair, which (by utilizing some random tie-breaking strategy) would make `JES` equivalent to TS. However, if `JES` is to be used in a completely noiseless setting, we argue that a small noise term should be added as a remedy. As this is done by default in many prominent GP frameworks [14, 13, 45], we do not view this as a major limitation of our approach. Nevertheless, improving upon this strategy would be interesting in future work.

For future work, we also envision work on the adaptation of `JES` to various different domains, such as multi-fidelity [48] and multi-objective optimization [1], as well as the integration of user prior knowledge over the location of the optimum [20] to accelerate optimization.

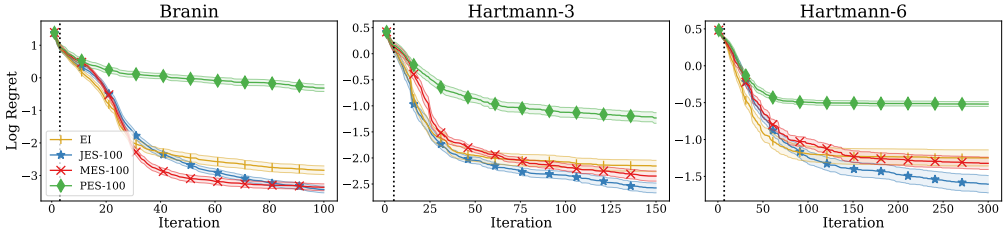

Fig. 2.7: Comparison of `JES`, `MES`, `PES` and `EI` on Branin and Hartmann-6, $\sigma_n^2 = 0.10$. Mean and 2 standard errors of log regret are displayed for each acquisition function across 100 repetitions. The vertical dashed line represents the end of the initial design phase.

Acknowledgements

Luigi Nardi was supported in part by affiliate members and other supporters of the Stanford DAWN project — Ant Financial, Facebook, Google, Intel, Microsoft, NEC, SAP, Teradata, and VMware. Carl Hvarfner and Luigi Nardi were partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Luigi Nardi was partially supported by the Wallenberg Launch Pad (WALP) grant Dnr 2021.0348. Frank Hutter acknowledges support through TAILOR, a project funded by the EU Horizon 2020 research and innovation programme under GA No 952215, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant number 417962828, by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG, and by the European Research Council (ERC) Consolidator Grant "Deep Learning 2.0" (grant no. 101045765). The computations were also enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at LUNARC partially funded by the Swedish Research Council through grant agreement no. 2018-05973. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the ERC. Neither the European Union nor the ERC can be held responsible for them.

Funded by
the European Union

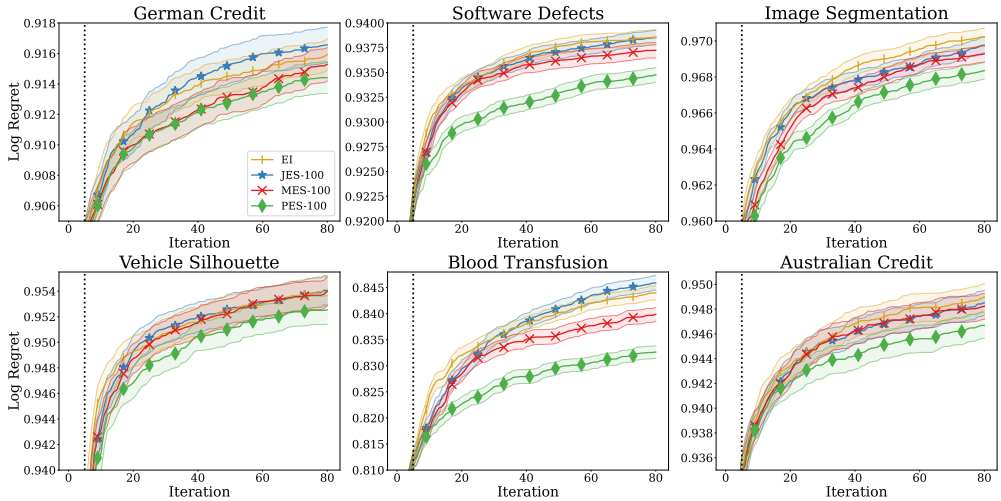

Fig. 2.8: Comparison of `JES`, `MES`, `PES` and `EI` on six different MLP tuning tasks from the HPOBench suite. Mean and 1 standard error of best observed accuracy are displayed for each acquisition function across 100 repetitions. The vertical dashed line represents the end of the initial design phase.

References

[1] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

[2] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24. Curran Associates, Inc., 2011.

[3] Felix Berkenkamp, Andreas Krause, and Angela Schoellig. Bayesian Optimization with Safety Constraints: Safe and Automatic Parameter Tuning in Robotics. *Machine Learning*, 06 2021. doi: 10.1007/s10994-021-06019-1.

[4] Salomon Bochner et al. *Lectures on Fourier integrals*, volume 42. Princeton University Press, 1959.

[5] Adam D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12:2879–2904, 2011.

[6] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. Deisenroth. Bayesian gait optimization for bipedal locomotion. In P. Pardalos and

M. Resende, editors, *Proceedings of the Eighth International Conference on Learning and Intelligent Optimization (LION'14)*, Lecture Notes in Computer Science. Springer, 2014.

[7] Yutian Chen, Aja Huang, Ziyu Wang, Ioannis Antonoglou, Julian Schrittwieser, David Silver, and Nando de Freitas. Bayesian optimization in alphago. *CoRR*, abs/1812.06855, 2018. URL `http://arxiv.org/abs/1812.06855`.

[8] Alexander Imani Cowen-Rivers, Wenlong Lyu, Zhi Wang, Rasul Tutunov, Jianye Hao, Jun Wang, and Haitham Bou-Ammar. HEBO: heteroscedastic evolutionary bayesian optimisation. *CoRR*, abs/2012.03826, 2020. URL `https://arxiv.org/abs/2012.03826`.

[9] George De Ath, Richard M. Everson, and Jonathan E. Fieldsend. Asynchronous ε-greedy bayesian optimisation. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 578–588, 27–30 Jul 2021.

[10] Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, Rene Sass, Aaron Klein, Noor Awad, Marius Lindauer, and Frank Hutter. HPOBench: A collection of reproducible multi-fidelity benchmark problems for HPO. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL `https://openreview.net/forum?id=1k4rJYEwda-`.

[11] Adel Ejjeh, Leon Medvinsky, Aaron Councilman, Hemang Nehra, Suraj Sharma, Vikram Adve, Luigi Nardi, Eriko Nurvitadhi, and Rob A Rutenbar. HPVM2FPGA: Enabling True Hardware-Agnostic FPGA Programming. In *Proceedings of the 33rd IEEE International Conference on Application-specific Systems, Architectures, and Processors*, 2022.

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

[13] Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

[14] GPy. GPy: A gaussian process framework in python. `http://github.com/SheffieldML/GPy`, since 2012.

[15] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(1):1809–1837, June 2012. ISSN 1532-4435.

[16] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, 2014. URL https://proceedings.neurips.cc/paper/2014/file/069d3bb002acd8d7dd095917f9efe4cb-Paper.pdf.

[17] José Miguel Hernández-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In *International conference on machine learning*, pages 1699–1707. PMLR, 2015.

[18] Matthew W. Hoffman and Zoubin Ghahramani. Output-space predictive entropy search for flexible global optimization. In *NeurIPS workshop on Bayesian optimization*, 2016.

[19] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*, 2011.

[20] Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. PiBO: Augmenting Acquisition Functions with User Beliefs for Bayesian Optimization. In *International Conference on Learning Representations*, 2022.

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

[22] H. J. Kushner. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, 86(1):97–106, 03 1964. ISSN 0021-9223. doi: 10.1115/1.3653121. URL https://doi.org/10.1115/1.3653121.

[23] Matthias Mayr, Faseeh Ahmad, Konstantinos I. Chatzilygeroudis, Luigi Nardi, and Volker Krüger. Skill-based Multi-objective Reinforcement Learning of Industrial Robot Tasks with Planning and Knowledge Integration. *CoRR*, abs/2203.10033, 2022. URL https://doi.org/10.48550/arXiv.2203.10033.

[24] Matthias Mayr, Carl Hvarfner, Konstantinos Chatzilygeroudis, Luigi Nardi, and Volker Krueger. Learning skill-based industrial robot tasks with user priors. *IEEE 18th International Conference on Automation Science and Engineering*, 2022. URL `https://arxiv.org/abs/2208.01605`.

[25] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

[26] Henry B. Moss, David S. Leslie, Javier Gonzalez, and Paul Rayson. Gibbon: General-purpose information-based bayesian optimisation. *Journal of Machine Learning Research*, 22(235):1–49, 2021. URL `http://jmlr.org/papers/v22/21-0120.html`.

[27] L. Nardi, D. Koeplinger, and K. Olukotun. Practical design space exploration. In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 347–358. IEEE, 2019.

[28] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Rectified max-value entropy search for bayesian optimization, 2022. URL `https://arxiv.org/abs/2202.13597`.

[29] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, Advances in Neural Information Processing Systems, page 1177–1184, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520.

[30] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[31] Binxin Ru, Michael A. Osborne, Mark Mcleod, and Diego Granziol. Fast information-theoretic Bayesian optimisation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4384–4392. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/ru18a.html`.

[32] Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. *Advances in Neural Information Processing Systems*, 27, 2014.

[33] Kenan Šehić, Alexandre Gramfort, Joseph Salmon, and Luigi Nardi. LassoBench: A High-Dimensional Hyperparameter Optimization Benchmark Suite for Lasso. *arXiv preprint arXiv:2111.02790*, 2021.

[34] Amar Shah and Zoubin Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. *Advances in neural information processing systems*, 28, 2015.

[35] B. Shahriari, K. Swersky, Z. Wang, R. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[36] Jack Sherman and Winifred J. Morrison. Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix. *The Annals of Mathematical Statistics*, 21(1):124 – 127, 1950. doi: 10.1214/aoms/1177729893. URL `https://doi.org/10.1214/aoms/1177729893`.

[37] J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian optimization of machine learning algorithms. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems (NeurIPS'12)*, pages 2960–2968, 2012.

[38] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, Prabhat, and R. Adams. Scalable Bayesian optimization using deep neural networks. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, volume 37, pages 2171–2180. Omnipress, 2015.

[39] J. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust Bayesian neural networks. In D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Proceedings of the 30th International Conference on Advances in Neural Information Processing Systems (NeurIPS'16)*, 2016.

[40] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, May 2012. ISSN 1557-9654. doi: 10.1109/tit.2011.2182033. URL `http://dx.doi.org/10.1109/TIT.2011.2182033`.

[41] Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9334–9345. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/takeno20a.html`.

[42] Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Sequential and parallel constrained max-value entropy search via information lower bound. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20960–20986. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/takeno22a.html`.

[43] W. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[44] Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective bayesian optimization. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=ZChgD8OoGds`.

[45] Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. Gpstuff: Bayesian modeling with gaussian processes. *Journal of Machine Learning Research*, 14:1175–1179, April 2013. ISSN 1532-4435.

[46] Julien Villemonteix, Emmanuel Vazquez, and Eric Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44, 12 2006. doi: 10.1007/s10898-008-9354-2.

[47] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning (ICML)*, 2017.

[48] Yehong Zhang, Trong Nghia Hoang, Bryan Kian Hsiang Low, and Mohan Kankanhalli. Information-based multi-fidelity bayesian optimization. In *NeurIPS Workshop on Bayesian Optimization*, 2017.

Paper III

Self-Correcting Bayesian Optimization through Bayesian Active Learning

Carl Hvarfner
Lund University

Erik Hellsten
Lund University

Frank Hutter
University of Freiburg

Luigi Nardi
Lund University

Abstract

Gaussian processes are the model of choice in Bayesian optimization and active learning. Yet, they are highly dependent on cleverly chosen hyperparameters to reach their full potential, and little effort is devoted to finding good hyperparameters in the literature. We demonstrate the impact of selecting good hyperparameters for GPs and present two acquisition functions that explicitly prioritize hyperparameter learning. Statistical distance-based Active Learning (`SAL`) considers the average disagreement between samples from the posterior, as measured by a statistical distance. `SAL` outperforms the state-of-the-art in Bayesian active learning on several test functions. We then introduce Self-Correcting Bayesian Optimization (`SCoreBO`), which extends `SAL` to perform Bayesian optimization and active learning simultaneously. `SCoreBO` learns the model hyperparameters at improved rates compared to vanilla BO, while outperforming the latest Bayesian optimization methods on traditional benchmarks. Moreover, we demonstrate the importance of self-correction on atypical Bayesian optimization tasks.

1 Introduction

Bayesian Optimization (BO) is a powerful paradigm for black-box optimization problems, i.e., problems that can only be accessed by pointwise queries. Such problems arise in many applications, ranging from including drug discovery [19] to configuration of combinatorial problem solvers [25, 26], hardware design [40, 13], hyperparameter tuning [31, 28, 48, 10], and robotics [8, 4, 37, 38].

Gaussian processes (GPs) are a popular choice as surrogate models in BO applications. Given the data, the model hyperparameters are typically estimated using either Maximum Likelihood or Maximum a Posteriori estimation (MAP) [45]. Alternatively, a fully Bayesian treatment of the hyperparameters [42, 50] removes the need to choose any single set through Monte Carlo integration. This procedure effectively considers all possible hyperparameter values under the current posterior, thereby accounting for hyperparameter uncertainty. However, the relationship between accurate GP hyperparameter estimation and BO performance has received little attention [65, 63, 3, 6, 53], and active reduction of hyperparameter uncertainty is not an integral part of any prevalent BO acquisition function. In contrast, the field of Bayesian Active Learning (BAL) contains multiple acquisition functions based solely on reducing hyperparameter-induced measures of uncertainty [24, 46, 32], and the broader field of Bayesian Experimental Design [9, 44, 1] revolves around acquisition of data to best learns the model parameters.

The importance of the GP hyperparameters in BO is illustrated in Fig. 3.1, which shows average simple regret over 20 optimization runs of 8-dimensional functions drawn from a Gaussian process prior. The curves correspond to the performance of Expected Improvement with noisy experiments (`EI`) [34] acquisition function under a fully Bayesian hyperparameter treatment using NUTS [23]. Two prevalent hyperparameter priors, as well as the true model hyperparameters, are used. Clearly, good model hyperparameters have substantial impact on BO performance, and BO methods could greatly benefit from estimating the model hyperparameters as accurately as possible. Furthermore, the hyperparameter estimation task can become daunting under complex problem setups, such as non-stationary objectives (spatially varying lengthscales, heteroskedasticity) [51, 12, 15, 5, 58], high-dimensional search spaces [14, 43], and additively decomposable objectives [30, 17]. The complexity of such problems warrants the use of more complex, task-specific surrogate models. In such settings, the success of the optimization may increasingly hinge on the presumed accuracy of the task-specific surrogate.

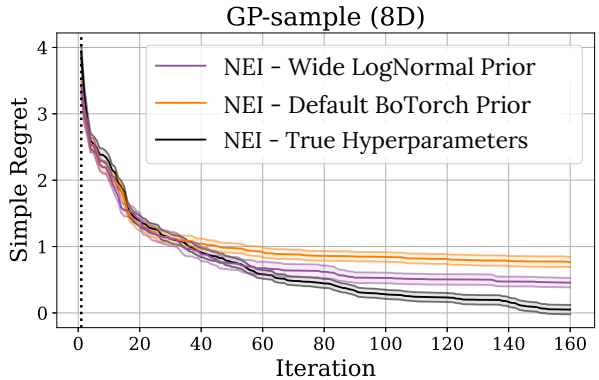

Fig. 3.1: Simple regret of using true hyperparameters, BoTorch (v.0.8.4 default) and lognormal hyperparameter priors with fully Bayesian hyperparameter treatment. The prior substantially impacts final performance, and correct hyperparameters yield vastly better results.

We proceed in two steps. We first introduce *Statistical distance-based Active Learning* (`SAL`), which improves Bayesian active learning by generalizing previous work [46, 24] and introduces a holistic measure of disagreement between the marginal posterior predictive distribution and each conditional posterior predictive. We consider the hyperparameter-induced disagreement between models in the acquisition function, thereby accelerating the learning of model hyperparameters. We then propose *Self-Correcting Bayesian Optimization* (`SCoreBO`), which builds upon `SAL` by explicitly learning the location of the optimizer in conjunction with model hyperparameters. This achieves accelerated hyperparameter learning and yields improved optimization performance on both conventional and exotic BO tasks. Formally, we make the following contributions:

1. We introduce `SAL`, a novel and efficient acquisition function for hyperparameter-oriented Bayesian active learning based on statistical distances (Sec. 3.1),
2. We introduce `SCoreBO`, the first acquisition function for joint BO and hyperparameter learning (Sec. 3.2),
3. We display highly competitive performance on an array of conventional AL (Sec. 4.1) and BO tasks (Sec. 4.2), and demonstrate `SCoreBOs` , ability to enhance atypical models such as SAASBO [14] and HEBO [12], and identify decompositions in AddGPs [30](Sec. 4.3).

2 Background

2.1 Gaussian processes

Gaussian processes (GPs) have become the model class of choice in most BO and active learning applications. They provide a distribution over functions $f \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$ fully defined by the mean function $m(\cdot)$ and the covariance function $k(\cdot,\cdot)$. Under this distribution, the value of the function $f(\boldsymbol{x})$, at a given point $\boldsymbol{x}$, is normally distributed with a closed-form solution for the mean and variance. We assume that observations are perturbed by Gaussian noise, such that $y_{\boldsymbol{x}} = f(\boldsymbol{x}) + \varepsilon,\ \varepsilon \sim N(0, \sigma_\varepsilon^2)$. We also assume the mean function to be constant, such that the dynamics are fully determined by the covariance function $k(\cdot,\cdot)$.

To account for differences in variable importance, each dimension is individually scaled using lengthscale hyperparameters ℓ_i. For D-dimensional inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$, the distance $r(\boldsymbol{x}, \boldsymbol{x}')$ is subsequently computed as $r^2 = \sum_{i=1}^{D}(x_i - x_i')^2/\ell_i^2$. Along with the outputscale σ_f, the set $\boldsymbol{\theta} = \{\boldsymbol{\ell}, \sigma_\varepsilon, \sigma_f\}$ comprises the set of hyperparameters that are conventionally learned. The likelihood surface for the GP hyperparameters is typically highly multi-modal [45, 64], where different modes represent different bias-variance trade-offs [45, 46]. To avoid having to choose a single mode, one can define a prior $p(\boldsymbol{\theta})$ and marginalize with respect to the hyperparameters when performing predictions [33].

2.2 Bayesian Optimization

Bayesian Optimization (BO) seeks to maximize to a black-box function f over a compact domain $\mathcal{X}$,

$$\boldsymbol{x}^* \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}), \tag{3.1}$$

such that f can only be sampled point-wise through expensive, noisy evaluations $y_{\boldsymbol{x}} = f(\boldsymbol{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$. New configurations are chosen by optimizing an *acquisition function*, which uses the surrogate model to quantify the utility of evaluating new points in the search space. Examples of such heuristics are Expected Improvement (`EI`) [29, 7] and Upper Confidence Bound (`UCB`) [52, 3, 55]. More sophisticated look-ahead approaches include Knowledge Gradient (`KG`) [16, 62] as well as a class of particular importance for our approach - the information-theoretic acquisition function class. These acquisition functions consider a mutual information objective to select the next query,

$$\alpha_{\texttt{MI}}(\boldsymbol{x}) = I(y_{\boldsymbol{x}}; \circledast \,|\mathcal{D}_n), \tag{3.2}$$

where $*$ can entail either the optimum $\boldsymbol{x}^*$ as in (Predictive) Entropy Search (ES/PES) [21, 22], the optimal value f^* as in Max-value Entropy Search (MES) [59, 54, 39] or the tuple $(\boldsymbol{x}^*, f^*)$, used in Joint Entropy Search (JES) [27, 56]. FITBO [47] shares similarities with our work, in that the optimal value is governed by a hyperparameter, in their case of a transformed GP.

Within BO, the fully Bayesian hyperparameter treatment is conventionally extended from the predictive posterior to the acquisition function such that for M models with hyperparameters $\boldsymbol{\theta}_m, m \in \{1, \dots, M\}$ sampled from the posterior over hyperparameters $p(\boldsymbol{\theta}|\mathcal{D})$, the acquisition function α is computed as an expectation over the hyperparameters [42, 50]

$$\alpha(\boldsymbol{x}|\mathcal{D}) = \mathbb{E}_{\boldsymbol{\theta}}[\alpha(\boldsymbol{x}|\boldsymbol{\theta}, \mathcal{D})] \approx \frac{1}{M} \sum_{m=1}^{M} \alpha(\boldsymbol{x}|\boldsymbol{\theta}_m, \mathcal{D}) \quad \boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D}). \tag{3.3}$$

This is also the definition of fully Bayesian treatment considered in this work.

2.3 Bayesian Active Learning

In contrast to BO, which aims to find a maximizer to an unknown function, Active Learning (AL) seeks to accurately learn the black-box function globally. Thus, the objective is to minimize the expected prediction loss. AL acquisition functions are classified as either *decision-theoretic*, which minimize the prediction loss over a validation set, or *information-theoretic*, which minimize the space of plausible models given the observed data [24, 35].

In the information-theoretic category, *Active Learning McKay* (ALM) [35] selects the point with the highest Shannon Entropy, which for GPs amounts to selecting the point with the highest variance. Under fully Bayesian hyperparameter treatment, it is referred to as Bayesian ALM (BALM). *Bayesian Active Learning by Disagreement* (BALD) [24] was among the first Bayesian active learning approaches to explicitly focus on learning the model hyperparameters. It approximates the reduction in entropy over the GP hyperparameters from observing a new data point

$$\alpha_{\texttt{BALD}}(\boldsymbol{x}) = I(y_{\boldsymbol{x}}; \boldsymbol{\theta}|\mathcal{D}) = \mathrm{H}(p(y_{\boldsymbol{x}}|\mathcal{D})) - \mathbb{E}_{\boldsymbol{\theta}}[\mathrm{H}(p(y_{\boldsymbol{x}}|\boldsymbol{\theta}, \mathcal{D}))] \tag{3.4}$$

and was later extended to deep Bayesian active learning [32] and active model (kernel) selection [18]. Lastly, [46] propose a *Bayesian Query-by-Committee* (BQBC) strategy. BQBC queries where the variance V of the GP mean is the

largest, with respect to changing model hyperparameters:

$$\alpha_{BQBC}(\boldsymbol{x}) = V_{\boldsymbol{\theta}}[\mu_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathcal{D})] = \mathbb{E}_{\boldsymbol{\theta}}[(\mu_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathcal{D}) - \mu(\boldsymbol{x}|\mathcal{D}))^2], \tag{3.5}$$

where $\mu(\boldsymbol{x})$ is the marginal posterior mean at $\boldsymbol{x}$, and $\mu_{\boldsymbol{\theta}}(\boldsymbol{x})$ is the posterior mean conditioned on $\boldsymbol{\theta}$. As such, `BQBC` queries the location which maximizes the average distance between the marginal posterior and the conditionals according to some distance metric (here, the posterior mean), henceforth referred to as hyperparameter-induced *posterior disagreement*. However, disagreement in mean alone does not fully capture hyperparameter-induced disagreement. Thus, [46] also presents *Query-by-Mixture of Gaussian Processes* (`QBMGP`), that adds the `BALM` criterion to the `BQBC` acquisition function.

2.4 Statistical Distances

A statistical distance quantifies the distance between two statistical objects. We focus on three (semi-)metrics, which have closed forms for Gaussian random variables.

The Hellinger distance is a dissimilarity measure between two probability distributions which has previously been employed in the context of BO-driven automated model selection by [36]. For two probability distributions p and q, it is defined as

$$H^2(p,q) = \frac{1}{2}\int_{\mathcal{X}} \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \lambda dx, \tag{3.6}$$

for some auxiliary measure λ under which both p and q are absolutely continuous.

The Wasserstein distance is dissimilarity metric between two distributions describing the average distance one distribution has to be moved to morph into another. The Wasserstein-k distance is defined as

$$W_k(p,q) = \left(\int_0^1 |F_q(x) - F_p(x)|^k dx\right)^{1/k} \tag{3.7}$$

where, in this work, we focus on the case where $k = 2$.

The KL divergence The KL divergence is a standard asymmetrical measure for dissimilarity between probability distributions. For two probability

distributions P and Q, it is given by $\mathcal{D}_{KL}(P \mid\mid Q) = \int_{\mathcal{X}} P(x)\log(P(x)/Q(x))dx$. The distances in Eq. (3.6), Eq. (3.7) and the KL divergence are used for the acquisition functions presented in Sec. 3.

3 Methodology

In Sec. 3.1, we introduce `SAL`, a novel family of metrics for BAL. In Sec. 3.2, we extend this to `SCoreBO`, the first acquisition function for joint BO and hyperparameter-oriented active learning, inspired by information-theoretic BO acquisition functions. In Sec. 3.3, we demonstrate how to efficiently approximate different types of statistical distances within the `SAL` context.

3.1 Statistical distance-based Active Learning

In active learning for GPs, it is important to efficiently learn the correct model hyperparameters. By measuring where the posterior hyperparameter uncertainty causes high disagreement in model output, the search can be focused on where this uncertainty has a high impact. However, considering only the posterior disagreement in mean, as in `BQBC`, is overly restrictive as it does not fully utilize the available distributions for the hyperparameters. For example, it ignores uncertainty in the outputscale hyperparameter of the Gaussian process, which disincentives exploration. As such, we propose to generalize the acquisition function in Eq. (3.5) to instead consider the posterior disagreement as measured by any statistical distance. Locations where the posterior distribution changes significantly as a result of model uncertainty are good points to query, in order to quickly learn the model hyperparameters. When an observation at such a location is obtained, hyperparameters which predicted that observation poorly will have a substantially smaller likelihood, which in turn aids hyperparameter convergence. The resulting `SAL` acquisition function is as follows:

$$\alpha_{SAL}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\theta}}[d(p(y_{\boldsymbol{x}}|\boldsymbol{\theta}, \mathcal{D}), p(y_{\boldsymbol{x}}|\mathcal{D}))] \approx \frac{1}{M}\sum_{m=1}^{M} d(p(y_{\boldsymbol{x}}|\boldsymbol{\theta}_m, \mathcal{D}), p(y_{\boldsymbol{x}}|\mathcal{D})), \tag{3.8}$$

where M is the number of hyperparameter samples drawn from its associated posterior, $\boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D})$, $\boldsymbol{\theta} = \{\boldsymbol{\ell}, \sigma_f, \sigma_\varepsilon\}$, and d is a statistical distance. Notably, SAL generalizes both BQBC and BALD, which are exactly recovered by choosing the semi-metric to the difference in mean or the forward KL divergence, respectively.

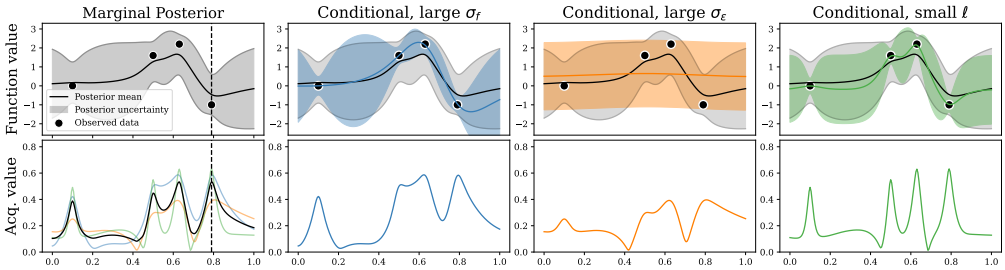

Fig. 3.2: Marginal posterior (top left, grey in other plots in top row), α_{SAL} using the Hellinger distance (bottom left, black), and the three conditional GPs (blue, orange, green) and their marginal contribution to the total acquisition function (bottom row). The large disagreement in noise level and lengthscale, primarily caused by the orange GP (large noise, long lengthscale), makes α_{SAL} query the lowest-valued point for a second time (selected location as vertical dashed line in the leftmost plot) to determine the mean and variance at that location.

Proposition 1. *SAL equipped with the KL-divergence is equivalent to BALD.*

Fig. 3.2 visualizes the SAL acquisition function. The marginal posterior (left) is made up of three vastly different conditional posteriors with hyperparameters sampled from $p(\boldsymbol{\theta}|\mathcal{D})$ - one with high outputscale (blue), one with very high noise (orange), and one with short lengthscale (green). For each of the blue, orange and green conditionals, the distance to the marginal posterior is computed. Intuitively, disagreement in noise level σ_ε can cause large posterior disagreement at already queried locations. Similarly, uncertainty in outputscale σ_f between posteriors will yield disagreement in large-variance regions, which will result in global variance reduction. Compared to other active learning acquisition functions, SAL carries distinct advantages: it has incentive to query the same location multiple times to estimate noise levels, and accomplishes the typical active learning objectives of predictive accuracy and global exploration by alleviating uncertainty over the lengthscales and outputscale of the GP. As we show in our experiments (Sec. 4.1,), SAL yields superior predictions and reduces hyperparameter uncertainty at drastically improved rates.

3.2 Self-Correcting Bayesian Optimization

Equipped with the SAL objective from Eq. (3.8), we have an intuitive measure for the hyperparameter-induced posterior disagreement, which incentivizes hyperparameter learning by querying locations where disagreement is the largest. However, it does not inherently carry an incentive to *optimize* the function. To inject an optimization objective into Eq. (3.8), we draw inspiration from

information-theoretic BO and further condition on samples of the optimum. Conditioning on potential optima yields an additional source of disagreement reserved for promising regions of the search space.

We consider $(\boldsymbol{x}^*, f^*)$, representing the global optimum and optimal value considered in JES [27, 56], as hyperparameters. When conditioning on $(\boldsymbol{x}^*, f^*)$, we condition on an additional observation, which displaces the mean and reduces the variance at $\boldsymbol{x}^*$. Moreover, the posterior over f becomes an upper truncated Gaussian, reducing the variance and pushing the mean marginally downwards in uncertain regions far away from the optimum as visualized in Fig. 3.3. Consequently, sampling and conditioning on $(\boldsymbol{x}^*, f^*)$ introduces an additional source of disagreement between the marginal posterior and the conditionals *globally*. The optimizer $(\boldsymbol{x}^*, f^*)$ is obtained through posterior sampling [61]. For brevity, we hereafter denote $(\boldsymbol{x}^*, f^*)$ by $*$. The resulting SCoreBO acquisition function is

$$\alpha_{SC}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\theta},*}[d(p(y_{\boldsymbol{x}}|\mathcal{D}), p(y_{\boldsymbol{x}}|\boldsymbol{\theta}, *, \mathcal{D}))]. \tag{3.9}$$

The joint posterior $p(\boldsymbol{\theta}, * |\mathcal{D}) = p(* |\boldsymbol{\theta}, \mathcal{D}) p(\boldsymbol{\theta}|\mathcal{D})$ used for the expectation in Eq. (3.9) can be approximated by hierarchical sampling. We first draw M hyperparameters $\boldsymbol{\theta}$ and thereafter N optimizers $* |\boldsymbol{\theta}$. As such, the expression for the SCoreBO acquisition function is:

$$\alpha(\boldsymbol{x}) \approx \frac{1}{NM} \sum_{m=1}^{M} \sum_{n=1}^{N} d\left(p(y_{\boldsymbol{x}}|\mathcal{D}), p(y_{\boldsymbol{x}}|\boldsymbol{\theta}_m, *_{\boldsymbol{\theta}_{m,n}}, \mathcal{D})\right), \tag{3.10}$$

where N is the number of optimizers sampled per hyperparameter set. Notably, while the acquisition function in (3.9) considers the optimizer $(\boldsymbol{x}^*, f^*)$, SCoreBO is not restricted to employing that quantity alone. Drawing parallels to PES and MES, we can also choose to condition on either $\boldsymbol{x}^*$ or f^* alone in place of $(\boldsymbol{x}^*, f^*)$. Doing so introduces a smaller disagreement in the posterior at the conditioned location $\boldsymbol{x}^*$, thus decreasing the acquisition value there. This will in turn decrease the emphasis that SCoreBO puts on optimization, relative to hyperparameter learning. In Fig. 3.3, the SCoreBO acquisition function is displayed for the same scenario as in Fig. 3.2. By conditioning on $N = 2$ optimizers per GP, we obtain $N \times M$ posteriors (displaying the posterior for one out of two optimizers, i.e. the left star in (blue), in Fig. 3.3). The mean is pushed upwards around the extra observation and the posterior predictive distribution over f is truncated as it is now upper bounded by f^*. While the preferred location under SAL is still attractive, the best location to query is now one that is more likely to be optimal, but still good under SAL.

Algorithm 3.1 displays how the involved densities are formed for one iteration of SCoreBO. For each hyperparameter set, a number of optima are sampled and

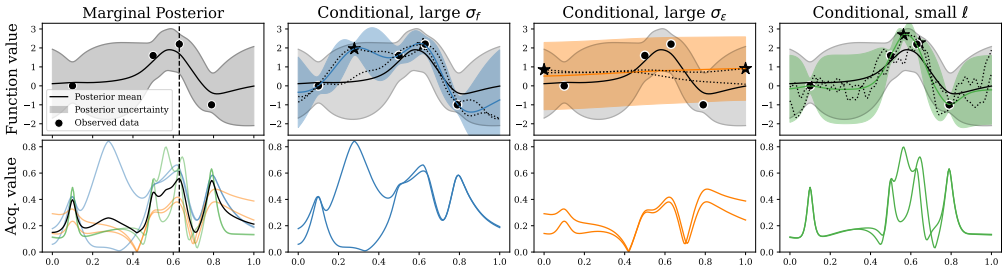

Fig. 3.3: Approximate marginal posterior after having conditioned on $(\boldsymbol{x}^*, f^*)$ (top left), α_{SC} using the Hellinger distance (bottom left), the three conditional truncated posteriors and their marginal contribution to the total acquisition function for the same iteration as Fig. 3.2. Conditioning on $(\boldsymbol{x}^*, f^*)$ (marked as $\star$, drawn from function samples in dashed) inroduces additional disagreement between the marginal posterior and the sampled GPs in promising regions as a result of conditioning. In the figure, we marginalize over $M = 3$ sets of hyperparameters and $N = 2$ optimizers per GP, where each optimizer's contribution to the acquisition function is visible under its corresponding GP. Note that, since function draws are *noiseless*, the conditioned optimum does not need to surpass the best *noisy* observation in value. This phenomenon is most notable in (orange).

individually conditioned on (`CondGP`) given the current data and hyperparameter set. After this procedure is completed for all hyperparameter sets, the statistical distance between each conditional posterior and the marginal is computed. The conditioning on the fantasized data point involves a rank-1 update of $\mathcal{O}(n^2)$ of the GP for each draw. As such, the complexity of constructing the acquisition functions is $\mathcal{O}(MNn^2)$ for M models, N optima per model and n data points. We utilize NUTS [23] for the MCMC involved with the fully Bayesian treatment, at a cost of $\mathcal{O}(Dn^3)$ per sample.

3.3 Approximation of Statistical Distances

We consider two proper statistical distances, Wasserstein distance and Hellinger distance. In contrast to `BQBC`, the statistical distance between the normally distributed conditionals and the marginal posterior predictive distribution (which is a Gaussian mixture), is not available in closed-form. We propose two approaches: estimating the distances using MC and estimation using moment matching (MM), which we outline below.

Approximation through Moment Matching We propose to fully utilize the closed-form expressions of the involved distances for Gaussians, and approximate the full posterior mixture $p(y_{\boldsymbol{x}}|\mathcal{D})$ with a Gaussian distribution using moment

Algorithm 3.1 `SCoreBO` iteration

Input: Number of hyperparameter sets M, number of sampled optima N, current data $\mathcal{D}$
Output: Next query location $\boldsymbol{x}'$.
for $m \in \{1, \dots, M\}$ **do**
  $\boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D})$
  for $n \in \{1, \dots, N\}$ **do**
    $*_{\boldsymbol{\theta}_{m,n}} \leftarrow \max f_{\boldsymbol{\theta}_{m,n}}$, where $f_{\boldsymbol{\theta}_{m,n}} \sim p(f|\boldsymbol{\theta}_m, \mathcal{D})$ {`Draw` n `optima for each` $\boldsymbol{\theta}_m$}
    $p(y_{\boldsymbol{x}}|\boldsymbol{\theta}_m, *_{\boldsymbol{\theta}_{m,n}}, \mathcal{D}) \leftarrow$ `CondGP`$(*_{\boldsymbol{\theta}_{m,n}}, \boldsymbol{\theta}_m, \mathcal{D})$ {`Condition GPs on each optimum`}
  end for
end for
$\boldsymbol{x}' = \arg\max \alpha(\boldsymbol{x})$ {`Defined in Eq.` (3.10)}

matching (MM) for the first and second moment. While a Gaussian mixture is not generally well approximated by a Normal distribution, the distance between the conditionals and the approximate posterior is small. In the moment matching approach, the conditional posterior $p(y_{\boldsymbol{x}}|\boldsymbol{\theta}, *, \mathcal{D})$ utilizes a lower bound on the change in the posterior induced by conditioning on $*$, as derived in GIBBON [39], which conveniently involves a second moment matching step of the extended skew Gaussian [41] $p(y_{\boldsymbol{x}}|\boldsymbol{\theta}, *, \mathcal{D})$. This naive approach circumvents a quadratic cost $\mathcal{O}(N^2M^2)$ in the number of samples of each pass through the acquisition function, and yields comparable performance to MC estimation.

4 Experiments

In this section we showcase the performance of the `SAL` and `SCoreBO` acquisition functions on a variety of tasks. For active learning, `SAL` shows state-of-the-art performance on a majority of benchmarks, and is more robust than the baselines. For the optimization tasks, `SCoreBO` more efficiently learns the model hyperparameters, and outperforms prominent Bayesian optimization acquisition functions on a variety of tasks. All experiments are implemented in BoTorch [2][3]. We use the same $\mathcal{LN}(0, 3)$[4] hyperparameter priors as [46] unless specified otherwise. `SCoreBO` *and all baselines* utilize fully Bayesian treatment of the hyperparameters. Our code is publicly available at `https://github.com/hvarfner/scorebo.git`. We utilize the moment matching approximation of the statistical distance.

[3] `https://botorch.org/` (v0.8.4)
[4] All Normal and LogNormal distributions are parametrized by the mean and *variance*.

4.1 Active Learning Tasks

To evaluate the performance of SAL, we compare it with BALD, BQBC and QBMGP on the same six functions used by [46]: Gramacy (1D) has a periodicity that is hard to distinguish from noise, Higdon and Gramacy (2D) varies in characteristics in different regions, whereas Branin, Hartmann-6 and Ishigami have a generally nonlinear structure. We display both the Wasserstein and Hellinger distance versions of SAL, denoted as SAL-WS and SAL-HR, respectively. We evaluate

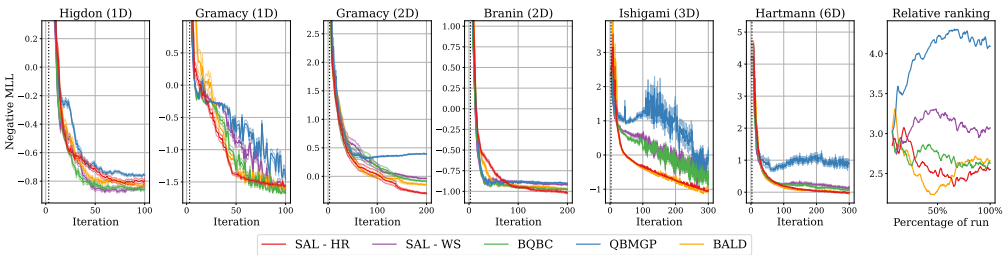

Fig. 3.4: Negative Marginal Log Likelihood (MLL) on six active learning functions and the (smoothed) relative rankings throughout each run for QBMGP, BQBC, BALD and SAL using Wasserstein and Hellinger distance. We plot mean and one standard error for 25 repetitions. SAL-HR is the top performing method, placing first in relative rankings. On Ishigami, only SAL-HR and BALD produces stable results.

each method on their predictive power, measured by the negative Marginal Log Likelihood (MLL) of the model predictions over a large set of validation points. MLL emphasizes calibration (accurate uncertainty estimates) in prediction over an accurate predictive mean. In Fig. 3.4, we show how the average validation set MLL changes with increasing training data. SAL-HR is the top-performing acquisition function on three out of six tasks, and rivals BALD for stability in predictive performance. This is particularly evident on the Ishigami function, where most methods fluctuate in the quality of their predictions. This can be attributed to emphasis on rapid hyperparameter learning In the rightmost plot, the real-time average per-seed ranking of acquisition function performance is displayed as a function of the fraction of budget expended. SAL-HR performs best, followed by BQBC andBALD. SAL-WS, however, does not display similarly consistent predictive quality as SAL-HR. The ability of SAL-HR to correctly estimate hyperparameters ensures calibrated uncertainty estimates, which makes it the better candidate for BO.

4.2 Bayesian Optimization Tasks

For the BO tasks, we use the Hellinger distance for its proficiency in prediction calibration and hyperparameter learning. We compare against several state-of-the-art baselines from the BO literature: `EI` for noisy experiments [34], as well as `JES` [27], the `MES` approach GIBBON [39] and PES [22]. As an additional reference, we include `EI` for noisy experiments [34] using MAP estimation.

Efficiently learning the hyperparameters To showcase `SCoreBO`'s ability to find the correct model hyperparameters, we run all relevant acquisition functions on samples from the 8-dimensional GP in Fig. 3.1. We exploit that for GP samples, the objectively true hyperparameters are known (in contrast to typical synthetic test functions). We utilize the same priors as in Fig. 3.1 on all the hyperparameters and compare `SCoreBO` to `EI` to assess the ability of each acquisition function to work independently of the choice of prior. In Fig. 3.5, for each acquisition function, we plot the average log regret over 20 dfifferent 8-dimensional instances of this task. The tasks at hand have lengthscales that vary substantially between dimensions.

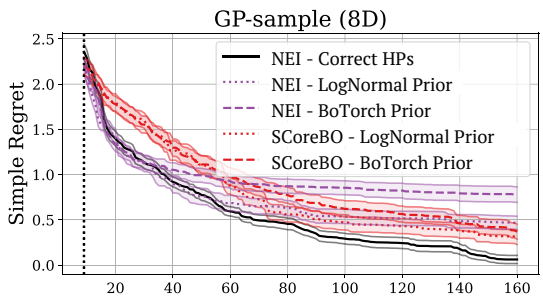

Fig. 3.5: Regret for `EI` and `SCoreBO` on the 8-dimensional GP sample for two different types of hyperparameter priors. Mean and standard deviation are plotted for all hyperparameter samples across 20 repetitions.

Synthetic test functions We run `SCoreBO` on a number of commonly used synthetic test functions for $25|\boldsymbol{\theta}|$ iterations, and present how the log inference regret evolves over the iterations in Fig. 3.6. All benchmarks are perturbed by Gaussian noise. We evaluate inference regret, i.e., the current best guess of the optimal location $\arg\max_{\boldsymbol{x}} \mu(\boldsymbol{x})$, which is conventional for non-myopic acquisition functions [20, 22, 27]. `SCoreBO` yields the the best final regret on four of the six tasks. In the relative rankings (rightmost plot), `SCoreBO` ranks poorly

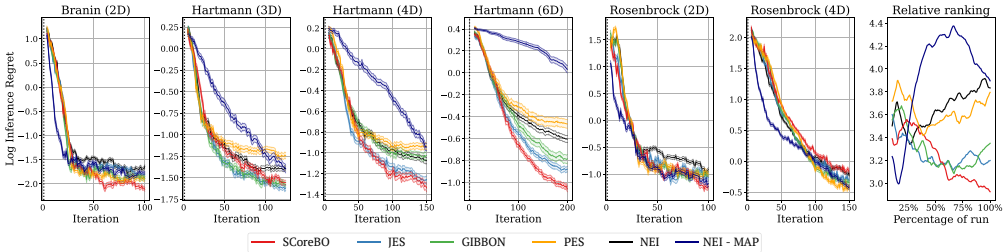

Fig. 3.6: Average log inference regret and (smoothed) relative ranking across 50 repetitions between the acquisition functions for `SCoreBO`, `JES`, `MES` and `EI` on six synthetic test functions. `SCoreBO` produces the best final regret on 4 out of 6 tasks, and has a substantially lower average ranking by the end of each run.

initially, but once hyperparameters are learned approximately halfway through the run, it substantially outperforms the competition. On Rosenbrock (4D), the relatively poor performance can explained by the apparent non-stationarity of the task , which makes hyperparameters diverge over time. This exposes a weakness of `SCoreBO`: When the modeling assumptions (such as stationarity) do not align with the task, optimization performance may suffer due to perpetual disagreement in the posterior.

4.3 A Practical Need for Self-correction

Lastly, we evaluate the performance of `SCoreBO` on three atypical tasks with increased emphasis on the surrogate model: (1) high-dimensional BO through sparse adaptive axis-aligned priors (SAASBO) [14], (2) BO with additively decomposable structure (AddGPs) [30, 17] and (3) non-stationary, heteroskedastic modelling with HEBO [12]. [14] consider their proposed method for noiseless tasks, where active variables easily distinguish from their non-active counterparts. However, SAASBO is not restricted to noiseless tasks. For AddGPs, data cross-covariance, and lack thereof, is similarly difficult to infer on noisy tasks.

In Fig. 3.7, we visualize the performance of `SCoreBO` and competing acquisition functions *with SAASBO priors* on two noisy benchmarks, Ackley-4 and Hartmann-6, with dummy dimensions added, as well as two real-world benchmarks: fitting a weighted Lasso model in 180 dimensions [49], and the tuning of all 385 lengthscales and three regularization parameters of an SVM [11], a task also considered by [14]. On these benchmarks, where finding the correct hyperparameters is crucial for performance, `SCoreBO` clearly outperforms traditional methods. To further exemplify how `SCoreBO` identifies the relevant dimensions, in Fig. 3.8, we show how the hyperparameters evolve on the 25D-embedded

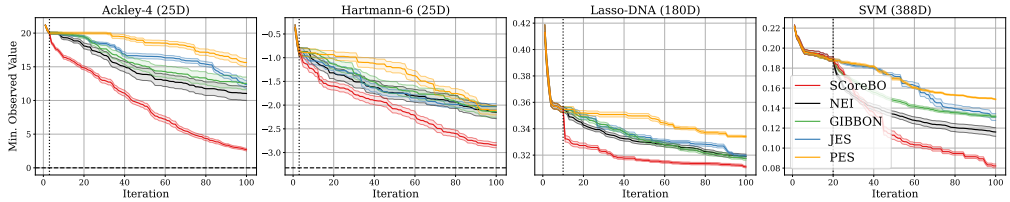

Fig. 3.7: Final loss using SAASBO priors on the noisy embedded Ackley-4, embedded Hartmann-6, the DNA classification and the SVM HPO task, mean and one standard error. `SCoreBO` identifies the important dimensions rapidly, and successfully optimizes the tasks. The optimal value is marked with a dashed line.

Ackley (4D) task. `SCoreBO` quickly finds the correct lengthscales and outputscale with high certainty, whereas `EI` remains uncertain of which dimensions are active throughout the optimization procedure. Impressively, `SCoreBO` finds accurate hyperparameters even faster than `BALD`, despite the latter being a pure active learning approach.

Secondly, we demonstrate the ability of `SCoreBO` to self-correct on *uncertainty in kernel design*, by considering AddGP tasks. We utilize the approach of [17], where additive decompositions are marginalized over. Ideally, a sufficiently accurate decomposition is found quickly, which rapidly speeds up optimization through accurate cross-correlation of data. Fig. 3.9 demonstrates `SCoreBO`'s performance on two GP sample tasks and a real-world task estimating cosmological constants (leftmost 3 plots) and its ability to find the correct additive decompositions (right). We observe that `SCoreBO` identifies correct decompositions substantially better than `EI`. Final performance, however, is only marginally better, as substantial resources are expended finding the right decompositions. Notably, the Cosmological Constants task does not display additive decomposability. As such, `SCoreBO` unsuccessfully expends resources attempting to reduce disagreement over additive structures, which hampers performance. This demonstrates that while `SCoreBO` learns the problem structure at increased rates, improved BO performance does not automatically follow.

Lastly, we apply `SCoreBO` to the `HEBO` [12] GP model, the winner of the NeurIPS 2020 Black-box optimization challenge [57]. The model employs input [51] and output warpings, the former of which are learnable to account for the heteroskedasticity that is prevalent in real-world optimization, and particularly HPO [51, 12], tasks. The complex model provides additional degrees of freedom in learning the objective. We evaluate `SCoreBO` and all baselines on three 4D deep learning HPO tasks: two involving large language models, and one from computer vision, from the PD1 [60] benchmarking suite. Fig. 3.10 displays that

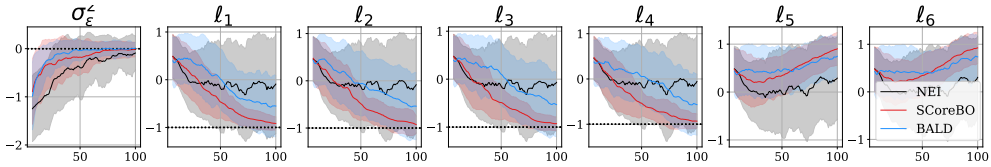

Fig. 3.8: Hyperparameter convergence on the $25D$-embedded $4D$ Ackley function with a SAASBO HP prior for `SCoreBO`, `EI` and `BALD`. Log HP mean and 1 standard deviation is plotted per iteration. `SCoreBO` identifies $\ell_1, \ldots, \ell_4$ as important (short lengthscales, $\ell_i \approx 10^{-1}$) with low uncertainty and $\ell_5, \ldots, \ell_{25}$ as dummy dimensions ($\ell_i \gg 10^1$). `EI` fails to identify any important lengthscales, whereas `SCoreBO` correctly identifies active dimensions with high certainty.. Notably, `SCoreBO` finds accurate hyperparameters even faster than `BALD`, a pure active learning approach. Reference HP values (where available) are marked with a dashed line.

`SCoreBO` obtains the best final accuracy on 2 out of 3 tasks, suggesting that self-correction is warranted for optimization of deep learning pipelines.

5 Conclusion and Future Work

The hyperparameters of Gaussian processes play an integral role in the efficiency of both Bayesian optimization and active learning applications. In this paper, we propose Statistical distance-based Active Learning (`SAL`) and Self-Correcting Bayesian Optimization (`SCoreBO`), two acquisition functions that explicitly consider hyperparameter-induced disagreement in the posterior distribution when selecting which points to query. We achieve high-end performance on both active learning and Bayesian optimization tasks, and successfully learn hyperparameters and kernel designs at improved rates compared to conventional methods. `SCoreBO` breaks ground for new methods in the space of joint active learning and optimization of black-box functions, which allows it to excel in high-dimensional BO, where learning important dimensions are vital. Moreover, the potential downside of self-correction is displayed when the model structure does not support the task at hand, or when self-correction is not required to solve the task. For future work, we will explore additional domains in which `SAL` and `SCoreBO` can allow for increased model complexity in BO applications.

6 Limitations

`SCoreBO` displays the ability to increase optimization efficiency on complex tasks that necessitate accurate modeling. However, `SCoreBO`'s efficiency is ultimately

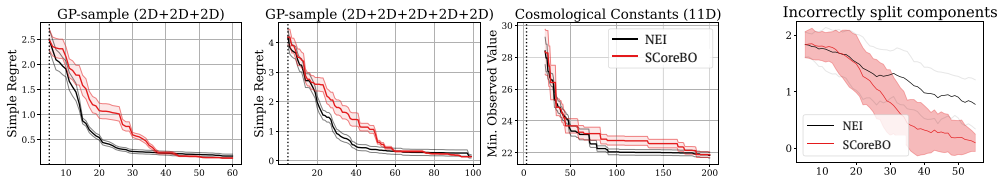

Fig. 3.9: Final value of using AddGPs on 6D and 10D GP sample functions, fully decomposable in groups of two, and the Cosmological Constants tasks. SCoreBO achieves better final performance (left, middle) with low uncertainty, and successfully finds the additive components of the 6D task (right).

contingent on the intrinsic ability of the GP to model the task at hand. this is demonstrated for the Rosenbrock (4D) function, where `SCoreBO` performs worse relative to other acquisition functions. There, the hyperparameter values increase over time instead of converge, which suggests that the objective is not part of the class of functions defined by the kernel. Thus, the self-correction effort is less helpful towards optimization. Moreover, increasing the model capacity, such as in Sec. 4.3, comes with increasing resources allocated towards self-correction. In highly constrained-budget applications, such resource allocation may not yield the best result, especially if increased model complexity is unwarranted. This is evident from the synthetic AddGP tasks, where despite accurately identifying the additive components, `SCoreBO` does not provide substantial performance gains over `EI`. Lastly, `SCoreBO`'s reliance on fully Bayesian hyperparameter treatment makes it more computationally demanding than MAP-based alternatives, limiting its use in high-throughput applications.

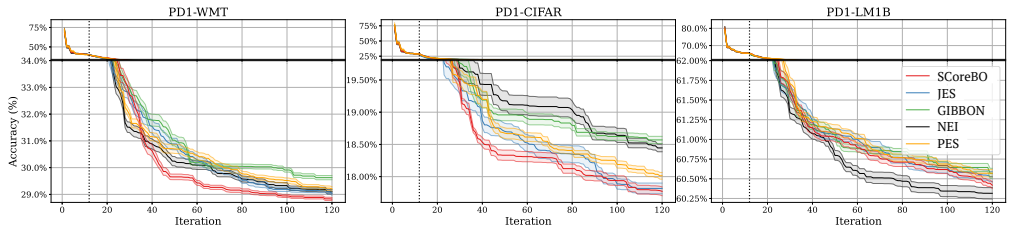

Fig. 3.10: Performance on the PD1 deep learning tasks over 20 repetitions using the warpings from HEBO [12]. `SCoreBO` obtains the best final accuracy on 2 out of 3 tasks, placing second on the third.

Acknowledgements

We thank the anonymous reviewers for their valuable contributions related `SAL`-KL and and its relationship with `BALD`, as well as their general feedback on the clarity of the paper and how our method was conveyed. We also thank Eytan Bakshy for the constructive feedback on earlier versions of this paper. Luigi Nardi was supported in part by affiliate members and other supporters of the Stanford DAWN project — Ant Financial, Facebook, Google, Intel, Microsoft, NEC, SAP, Teradata, and VMware. Carl Hvarfner, Erik Hellsten and Luigi Nardi were partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Luigi Nardi was partially supported by the Wallenberg Launch Pad (WALP) grant Dnr 2021.0348. Frank Hutter acknowledges support through TAILOR, a project funded by the EU Horizon 2020 research and innovation programme under GA No 952215, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant number 417962828, by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG, and by the European Research Council (ERC) Consolidator Grant "Deep Learning 2.0" (grant no. 101045765). The computations were also enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at LUNARC partially funded by the Swedish Research Council through grant agreement no. 2018-05973. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the ERC. Neither the European Union nor the ERC can be held responsible for them.

Funded by the European Union

References

[1] A.C. Atkinson and A.N. Donev. *Optimum Experimental Designs*. Oxford science publications. Clarendon Press, 1992. ISBN 9780198522546. URL https://books.google.se/books?id=cmmOA_-M7S0C.

[2] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. In *Advances in Neural Information Processing Systems*, 2020. URL http://arxiv.org/abs/1910.06403.

[3] Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. No-regret bayesian optimization with unknown hyperparameters. *Journal of Machine Learning Research*, 20(50):1–24, 2019. URL http://jmlr.org/papers/v20/18-213.html.

[4] Felix Berkenkamp, Andreas Krause, and Angela Schoellig. Bayesian Optimization with Safety Constraints: Safe and Automatic Parameter Tuning in Robotics. *Machine Learning*, 06 2021. doi: 10.1007/s10994-021-06019-1.

[5] Erik Bodin, Markus Kaiser, Ieva Kazlauskaite, Zhenwen Dai, Neill Campbell, and Carl Henrik Ek. Modulating surrogates for Bayesian optimization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 970–979. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/bodin20a.html.

[6] Ilija Bogunovic and Andreas Krause. Misspecified gaussian process bandit optimization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3004–3015. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/177db6acfe388526a4c7bff88e1feb15-Paper.pdf.

[7] Adam D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12:2879–2904, 2011.

[8] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. Deisenroth. Bayesian gait optimization for bipedal locomotion. In P. Pardalos and M. Resende, editors, *Proceedings of the Eighth International Conference on Learning and Intelligent Optimization (LION'14)*, Lecture Notes in Computer Science. Springer, 2014.

[9] Kathryn Chaloner and Isabella Verdinelli. Bayesian Experimental Design: A Review. *Statistical Science*, 10(3):273 – 304, 1995. doi: 10.1214/ss/1177009939. URL https://doi.org/10.1214/ss/1177009939.

[10] Yutian Chen, Aja Huang, Ziyu Wang, Ioannis Antonoglou, Julian Schrittwieser, David Silver, and Nando de Freitas. Bayesian optimization in alphago. *CoRR*, abs/1812.06855, 2018. URL http://arxiv.org/abs/1812.06855.

[11] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20: 273–297, 1995.

[12] Alexander Imani Cowen-Rivers, Wenlong Lyu, Zhi Wang, Rasul Tutunov, Jianye Hao, Jun Wang, and Haitham Bou-Ammar. HEBO: heteroscedastic evolutionary bayesian optimisation. *CoRR*, abs/2012.03826, 2020. URL https://arxiv.org/abs/2012.03826.

[13] Adel Ejjeh, Leon Medvinsky, Aaron Councilman, Hemang Nehra, Suraj Sharma, Vikram Adve, Luigi Nardi, Eriko Nurvitadhi, and Rob A Rutenbar. HPVM2FPGA: Enabling True Hardware-Agnostic FPGA Programming. In *Proceedings of the 33rd IEEE International Conference on Application-specific Systems, Architectures, and Processors*, 2022.

[14] David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 493–503. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/eriksson21a.html.

[15] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/6c990b7aca7bc7058f5e98ea909e924b-Paper.pdf.

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

[17] J. Gardner, C. Guo, K. Weinberger, R. Garnett, and R. Grosse. Discovering and Exploiting Additive Structure for Bayesian Optimization. In A. Singh and J. Zhu, editors, *Proceedings of the Seventeenth International Conference*

on Artificial Intelligence and Statistics (AISTATS), volume 54, pages 1311–1319. Proceedings of Machine Learning Research, 2017.

[18] Jacob Gardner, Gustavo Malkomes, Roman Garnett, Kilian Q Weinberger, Dennis Barbour, and John P Cunningham. Bayesian active model selection with an application to automated audiometry. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper_files/paper/2015/file/d9731321ef4e063ebbee79298fa36f56-Paper.pdf`.

[19] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design. *arXiv: Machine Learning*, 2017.

[20] P. Hennig and C. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 98888(1):1809–1837, 2012.

[21] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(1):1809–1837, June 2012. ISSN 1532-4435.

[22] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, 2014. URL `https://proceedings.neurips.cc/paper/2014/file/069d3bb002acd8d7dd095917f9efe4cb-Paper.pdf`.

[23] Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014. URL `http://jmlr.org/papers/v15/hoffman14a.html`.

[24] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[25] F. Hutter, H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In C. Coello, editor, *Proceedings of the Fifth International Conference on Learning and Intelligent Optimization (LION'11)*, volume 6683 of *Lecture Notes in Computer Science*, pages 507–523. Springer, 2011.

[26] F. Hutter, M. Lindauer, A. Balint, S. Bayless, H. Hoos, and K. Leyton-Brown. The configurable SAT solver challenge (CSSC). *Artificial Intelligence*, 243: 1–25, 2017.

[27] Carl Hvarfner, Frank Hutter, and Luigi Nardi. Joint entropy eearch for maximally-informed bayesian optimization. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022.

[28] Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. PiBO: Augmenting Acquisition Functions with User Beliefs for Bayesian Optimization. In *International Conference on Learning Representations*, 2022.

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

[30] K. Kandasamy, J. Schneider, and B. Póczos. High Dimensional Bayesian Optimisation and Bandits via Additive Models. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, volume 37, pages 295–304. Omnipress, 2015.

[31] Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Neural architecture search with bayesian optimisation and optimal transport. *Advances in neural information processing systems*, 31, 2018.

[32] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019.

[33] Vidhi Lalchand and Carl Edward Rasmussen. Approximate inference for fully bayesian gaussian process regression. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–12. PMLR, 2020.

[34] B. Letham, K. Brian, G. Ottoni, and E. Bakshy. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 2018.

[35] David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

[36] Gustavo Malkomes, Chip Schaff, and Roman Garnett. Bayesian optimization for automated model selection. In Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors, *Proceedings of the Workshop on Automatic Machine*

Learning, volume 64 of *Proceedings of Machine Learning Research*, pages 41–47, New York, New York, USA, 24 Jun 2016. PMLR. URL https://proceedings.mlr.press/v64/malkomes_bayesian_2016.html.

[37] Matthias Mayr, Faseeh Ahmad, Konstantinos I. Chatzilygeroudis, Luigi Nardi, and Volker Krüger. Skill-based Multi-objective Reinforcement Learning of Industrial Robot Tasks with Planning and Knowledge Integration. *CoRR*, abs/2203.10033, 2022. URL https://doi.org/10.48550/arXiv.2203.10033.

[38] Matthias Mayr, Carl Hvarfner, Konstantinos Chatzilygeroudis, Luigi Nardi, and Volker Krueger. Learning skill-based industrial robot tasks with user priors. *IEEE 18th International Conference on Automation Science and Engineering*, 2022. URL https://arxiv.org/abs/2208.01605.

[39] Henry B. Moss, David S. Leslie, Javier Gonzalez, and Paul Rayson. Gibbon: General-purpose information-based bayesian optimisation. *Journal of Machine Learning Research*, 22(235):1–49, 2021. URL http://jmlr.org/papers/v22/21-0120.html.

[40] L. Nardi, D. Koeplinger, and K. Olukotun. Practical design space exploration. In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 347–358. IEEE, 2019.

[41] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Rectified max-value entropy search for bayesian optimization, 2022. URL https://arxiv.org/abs/2202.13597.

[42] Michael A Osborne. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, Oxford University, UK, 2010.

[43] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Increasing the scope as you learn: Adaptive bayesian optimization in nested subspaces. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=e4Wf6112DI.

[44] Tom Rainforth, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. Modern bayesian experimental design, 2023.

[45] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[46] Christoffer Riis, Francisco N Antunes, Frederik Boe Hüttel, Carlos Lima Azevedo, and Francisco Camara Pereira. Bayesian active learning with fully bayesian gaussian processes. *arXiv preprint arXiv:2205.10186*, 2022.

[47] Binxin Ru, Michael A. Osborne, Mark Mcleod, and Diego Granziol. Fast information-theoretic Bayesian optimisation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4384–4392. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/ru18a.html`.

[48] Binxin Ru, Xingchen Wan, Xiaowen Dong, and Michael Osborne. Interpretable neural architecture search via bayesian optimisation with weisfeiler-lehman kernels. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=j9Rv7qdXjd`.

[49] Kenan Šehić, Alexandre Gramfort, Joseph Salmon, and Luigi Nardi. LassoBench: A High-Dimensional Hyperparameter Optimization Benchmark Suite for Lasso. *arXiv preprint arXiv:2111.02790*, 2021.

[50] J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian optimization of machine learning algorithms. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems (NeurIPS'12)*, pages 2960–2968, 2012.

[51] J. Snoek, K. Swersky, R. Zemel, and R. Adams. Input warping for Bayesian optimization of non-stationary functions. In E. Xing and T. Jebara, editors, *Proceedings of the 31th International Conference on Machine Learning, (ICML'14)*, pages 1674–1682. Omnipress, 2014.

[52] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, May 2012. ISSN 1557-9654. doi: 10.1109/tit.2011.2182033. URL `http://dx.doi.org/10.1109/TIT.2011.2182033`.

[53] Samuel Stanton, Wesley Maddox, and Andrew Gordon Wilson. Bayesian optimization with conformal prediction sets. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 959–986. PMLR, 25–27 Apr 2023. URL `https://proceedings.mlr.press/v206/stanton23a.html`.

[54] Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9334–9345. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/takeno20a.html`.

[55] Shion Takeno, Yu Inatsu, and Masayuki Karasuyama. Randomized Gaussian process upper confidence bound with tighter Bayesian regret bounds. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 33490–33515. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/takeno23a.html`.

[56] Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective bayesian optimization. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=ZChgD8OoGds`.

[57] Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In Hugo Jair Escalante and Katja Hofmann, editors, *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, pages 3–26. PMLR, 06–12 Dec 2021. URL `https://proceedings.mlr.press/v133/turner21a.html`.

[58] Xingchen Wan, Vu Nguyen, Huong Ha, Binxin Ru, Cong Lu, and Michael A Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. *International Conference on Machine Learning (ICML) 38*, 2021.

[59] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning (ICML)*, 2017.

[60] Zi Wang, George E. Dahl, Kevin Swersky, Chansoo Lee, Zachary Nado, Justin Gilmer, Jasper Snoek, and Zoubin Ghahramani. Pre-trained Gaussian processes for Bayesian optimization. *arXiv preprint arXiv:2109.08215*, 2023.

[61] James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Efficiently sampling functions from gaussian process posteriors. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/2002.09309.

[62] Jian Wu, Matthias Poloczek, Andrew G Wilson, and Peter Frazier. Bayesian optimization with gradients. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/64a08e5f1e6c39faeb90108c430eb120-Paper.pdf.

[63] George Wynne, François-Xavier Briol, and Mark Girolami. Convergence guarantees for gaussian process means with misspecified likelihoods and smoothness. *J. Mach. Learn. Res.*, 22(1), jan 2021. ISSN 1532-4435.

[64] Yuling Yao, Aki Vehtari, and Andrew Gelman. Stacking for non-mixing bayesian computations: The curse and blessing of multimodal posteriors, 2020.

[65] Boya Zhang, D. Austin Cole, and Robert B. Gramacy. Distance-distributed design for gaussian process surrogates, 2019.

Paper IV

A General Framework for User-Guided Bayesian Optimization

Carl Hvarfner
Lund University

Frank Hutter
University of Freiburg

Luigi Nardi
Lund University

Abstract

The optimization of expensive-to-evaluate black-box functions is prevalent in various scientific disciplines. Bayesian optimization is an automatic, general and sample-efficient method to solve these problems with minimal knowledge of the underlying function dynamics. However, the ability of Bayesian optimization to incorporate prior knowledge or beliefs about the function at hand in order to accelerate the optimization is limited, which reduces its appeal for knowledgeable practitioners with tight budgets. To allow domain experts to customize the optimization routine, we propose `ColaBO`, the first Bayesian-principled framework for incorporating prior beliefs beyond the typical kernel structure, such as the likely location of the optimizer or the optimal value. The generality of `ColaBO` makes it applicable across different Monte Carlo acquisition functions and types of user beliefs. We empirically demonstrate `ColaBO`'s ability to substantially accelerate optimization when the prior information is accurate, and to retain approximately default performance when it is misleading.

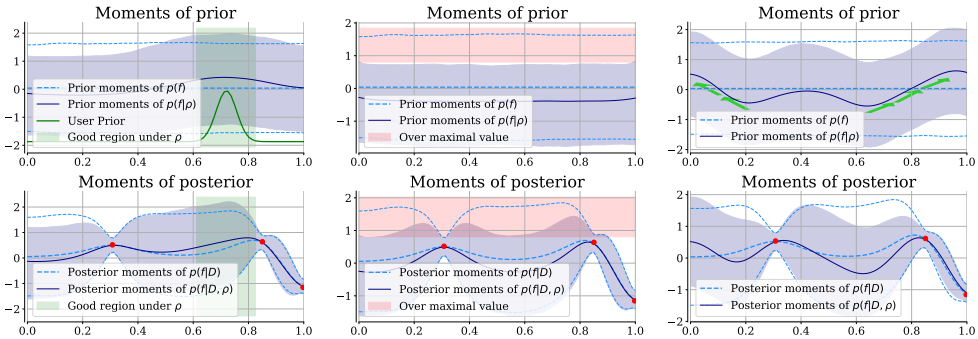

Fig. 4.1: Three different `ColaBO` priors: (left) Prior over the optimum $\boldsymbol{x}^*$, and the induced changed in the GP for an optimum located in the green region. (middle) Prior over optimal value, $f^* < 0.8$. (right) Prior over preference relations $f(\boldsymbol{x})_1 \succcurlyeq f(\boldsymbol{x}_2)$ for five preferences (green arrows, e.g. $f(0.0) \succcurlyeq f(0.1) \succcurlyeq f(0.2)$.

1 Introduction

Bayesian Optimization (BO) [39, 27, 57] is a well-established methodology for the optimization of expensive-to-evaluate black-box functions. Known for its sample efficiency, BO has been successfully applied to a variety of domains where laborious system tuning is prominent, such as hyperparameter optimization [57, 5, 36], neural architecture search [54, 70], robotics [8, 38], hardware design [44, 9], and chemistry [17].

Typically employing a Gaussian Process [50] (GP) surrogate model, BO allows the user to specify a prior over functions $p(f)$ via the Gaussian Process kernel, as well as an optional prior over its hyperparameters. Within the framework of the prior, the user can specify expected smoothness, output range and possible noise level of the function at hand, with the hopes of accelerating the optimization if accurate. However, the prior beliefs that can be specified within the framework of the kernel hyperparameters do not span the full range of beliefs that practitioners may possess. For example, practitioners may know which *parts of the input space* tend to work best [47, 48, 56, 66], know a range or upper bound on the optimal output [26, 46] such as a maximal achievable accuracy of 100%, or other properties of the objective, such as preference relations between configurations [21]. The limited ability of practitioners to interact and collaborate with the BO machinery [32] thus runs the risk of failing to use valuable domain expertise, or alienating knowledgeable practitioners altogether. While knowledge injection beyond what is natively supported by the GP kernel is crucial to further increase the efficiency of Bayesian optimization, so far no current approach allows for the integration of arbitrary types of user knowledge. To address this gap, we

propose an intuitive framework that effectively allows the user to reshape the Gaussian process at will to mimic their held beliefs.

Figure 4.1 illustrates how, for the three aforementioned priors, the GP is reshaped to *faithfully represent* the belief held by the user - whether it be a prior over the optimum (left), optimal value (middle), or preference relations between points (right). Our novel framework for *Collaborative Bayesian Optimization* (`ColaBO`) diverges from the typical assumption of Gaussian posteriors, and is applicable to any Monte Carlo acquisition function [72, 71, 3]. Formally, we make the following contributions:

1. We introduce `ColaBO`, a framework for incorporating user knowledge over the optimizer, optimal value and preference relations into Bayesian optimization in the form of an additional prior on the surrogate, orthogonal to the conventional prior on the kernel hyperparameters,
2. We demonstrate that the proposed framework is generally applicable to Monte Carlo acquisition functions, inheriting MC acquisiton function utility,
3. We empirically show that `ColaBO` accelerates optimization when injected with for priors over optimal location and optimal value.

2 Background

We outline Bayesian optimization, Gaussian Processes and Monte Carlo (MC) acquisition functions, as well as the concept of a prior over the optimum.

2.1 Bayesian optimization

We consider the problem of optimizing a black-box function f across a set of feasible inputs $\mathcal{X} \subset \mathbb{R}^d$:

$$\boldsymbol{x}^* \in \arg\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}). \tag{4.1}$$

We assume that $f(\boldsymbol{x})$ is expensive to evaluate and can potentially only be observed through a noise-corrupted estimate, $y_{\boldsymbol{x}}$, where $y_{\boldsymbol{x}} = f(\boldsymbol{x}) + \varepsilon, \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ for some noise level σ_ε^2. In this setting, we wish to maximize f in an efficient manner. Bayesian optimization (BO) aims to globally maximize f by an initial design and thereafter sequentially choosing new points $\boldsymbol{x}_n$ for some iteration n, creating

the data $\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{(\boldsymbol{x}_n, y_n)\}$ [6, 55, 16]. After each new observation, BO constructs a probabilistic surrogate model $p(f|\mathcal{D}_n)$ [57, 22, 4, 42] and uses that surrogate to build an acquisition function $\alpha(\boldsymbol{x}; \mathcal{D}_n)$ which selects the next query.

2.2 Gaussian processes

When constructing the surrogate, the most common choice is a *Gaussian process* (GP) [50]. The GP utilizes a covariance function k, which encodes a prior belief for the smoothness of f, and determines how previous observations influence prediction. Given observations $\mathcal{D}_n$ at iteration n, the Gaussian posterior $p(f|\mathcal{D}_n)$ over the objective is characterized by the posterior mean $\mu_n(\boldsymbol{x}, \boldsymbol{x}')$ and (co-)variance $\Sigma_n(\boldsymbol{x}, \boldsymbol{x}')$ of the GP:

$$\begin{aligned}\mu_n(\boldsymbol{x}) &= \mathbf{k}_n(\boldsymbol{x})^\top (\mathbf{K}_n + \sigma_\varepsilon^2 \mathbf{I})^{-1} \mathbf{y} \\ \Sigma_n(\boldsymbol{x}, \boldsymbol{x}') &= k(\boldsymbol{x}, \boldsymbol{x}') - \mathbf{k}_n(\boldsymbol{x})^\top (\mathbf{K} + \sigma_\varepsilon^2 \mathbf{I})^{-1} \mathbf{k}_n(\boldsymbol{x}'),\end{aligned}$$

where $(\mathbf{K}_n)_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\mathbf{k}_n(\boldsymbol{x}) = [k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_n)]^\top$ and σ_ε^2 is the noise variance.

For applications in BO and beyond, samples from the posterior are required either directly for optimization [10] through Thompson sampling [63], or to estimate auxiliary quantities of interest [20, 45, 25]. For a finite set of k query locations $(\boldsymbol{X} = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$, the classical method of generating samples is via a location-scale transform of Gaussian random variables, $f(\boldsymbol{X}) = \mu_n(\boldsymbol{X}) + \boldsymbol{L}\boldsymbol{\varepsilon}$, where $\boldsymbol{L}$ is the Cholesky decomposition of $\boldsymbol{K}$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{I})$. Unfortunately, the classic approach is intrinsically non-scalable, incurring a $\mathcal{O}(k^3)$ cost due to the aforementioned matrix decomposition.

2.3 Decoupled Posterior Sampling

To remedy the issue of scalability in posterior sampling, $\mathcal{O}(k)$ weight-space approximations based on Random Fourier Features (RFF) [49] obtain approximate (continuous) function draws $\hat{f}(\boldsymbol{x}) = \sum_{i=1}^m w_i \phi_i(\boldsymbol{x})$, where $\phi_i(\boldsymbol{x}) = \frac{2}{\ell}(\boldsymbol{\psi}_i^\top \boldsymbol{x} + b_i)$. The random variables $w_i \sim \mathcal{N}(0, 1)$, $b_i \sim \mathcal{U}(0, 2\pi)$, and $\boldsymbol{\psi}_i$ are sampled proportional to the spectral density of k.

While achieving scalability, the seminal RFF approach by [49] suffers from the issue of variance starvation [43, 68, 73]. As a remedy, [73] decouple the draw of functions from the approximate posterior $p(\hat{f}|\mathcal{D})$ into a more accurate draw

from the prior $p(\hat{f})$, followed by a deterministic data-dependent update:

$$(\hat{f}|\mathcal{D})(\boldsymbol{x}) \overset{d}{=} \underbrace{\hat{f}(\boldsymbol{x})}_{\text{draw from prior}} + \underbrace{\mathbf{k}_n(\boldsymbol{x})^\top(\mathbf{K}_n + \sigma_\varepsilon^2\mathbf{I})^{-1}(\mathbf{y} - \hat{f}(\boldsymbol{x}) - \boldsymbol{\varepsilon})}_{\text{deterministic update}} \tag{4.2}$$

Eq. 4.2 deviates from the distribution-first approach that is typically prevalent in GPs in favor of a variable-first approach utilizing Matheron's rule [28].

2.4 Monte Carlo Acquisition Functions

Acquisition functions act on the surrogate model to quantify the utility of a point in the search space. They encode a trade-off between exploration and exploitation, typically using a greedy heuristic to do so. A simple and computationally cheap heuristic is Expected Improvement (`EI`) [27, 7]. For a noiseless function and a current best observation y_n^*, the `EI` acquisition function is $\alpha_{EI}(\boldsymbol{x}) = \mathbb{E}_{y_{\boldsymbol{x}}}[(y_n^* - y_{\boldsymbol{x}})^+]$. For noisy problem settings, a noise-adapted variant of `EI` [34] is frequently considered, where both the incumbent y_n^* and the upcoming query $y_{\boldsymbol{x}}$ are substituted for the non-observable noiseless incumbent f_n^* and noiseless upcoming query $f_{\boldsymbol{x}}$. Other frequently used acquisition functions are the Upper Confidence Bound (`UCB`) [59], Probability of Improvement (PI) [33] and Knowledge Gradient (KG) [14, 15]. Information-theoretic acquisition functions consider the mutual information to select the next query $\alpha_{\texttt{MI}}(\boldsymbol{x}) = I((\boldsymbol{x}, y_{\boldsymbol{x}}); *|\mathcal{D}_n)$, where $*$ can entail either the optimum $\boldsymbol{x}^*$ as in (Predictive) Entropy Search (`ES`/`PES`) [18, 19], the optimal value f^* as in Max-value Entropy Search (`MES`) [67, 40] or the tuple $(\boldsymbol{x}^*, f^*)$ for Joint Entropy Search (`JES`) [23, 65].

All the aforementioned acquisition functions compute expectations $\mathbb{E}_{f_{\boldsymbol{x}}}$ (or alternatively $\mathbb{E}_{y_{\boldsymbol{x}}}$ over some utility $u(f_{\boldsymbol{x}})$ of the output [72, 71], which typically have simple, or even closed-form, solutions for Gaussian posteriors. However, approximating the expectation through Monte Carlo integration has proven useful in the context of batch optimization [71], efficient acquisition function approximation [3], and non-Gaussian posteriors [2]. By sampling over possible outputs $f_{\boldsymbol{x}}$ and utilizing the reparametrization trick [31, 51], utilities u can be easily computed across a larger set of applications and be optimized to greater accuracy.

2.5 Prior over the Optimum

A prior over the optimum [58, 24, 37] is a user-specified belief $\pi : \mathcal{X} \to \mathbb{R}$ of the subjective likelihood that a given $\boldsymbol{x}$ is optimal. Formally,

$$\pi(\boldsymbol{x}) = \mathbb{P}\left(\boldsymbol{x} = \arg\max_{\boldsymbol{x}'} f(\boldsymbol{x}')\right). \tag{4.3}$$

This prior is generally considered to be independent of observed data, but rather a result of previous experimentation or anecdotal evidence. Regions that the user expects to contain the optimum will typically have a high value, but this does not exclude the chance of the user belief $\pi(\boldsymbol{x})$ to be inaccurate, or even misleading. Lastly, we require π to be strictly positive in all of $\mathcal{X}$, which suggests that any point included in the search space may be optimal.

3 Methodology

We now introduce `ColaBO`, the first Bayesian-principled BO framework that flexibly allows users to *collaborate* with the optimizer by injecting prior knowledge about the objective that substantially exceeds the type of prior knowledge natively supported by GPs. In Sec. 3.1, we introduce and derive a novel prior over function properties, which yields a surrogate model conditioned on the user belief. Thereafter, in Sec. 3.2, we demonstrate how the hierarchical prior integrates with MC acquisition functions. Lastly, in Sec. 3.3, we state practical considerations to assure the performance of `ColaBO`.

3.1 Prior over Function Properties

We consider the typical GP prior over functions $p(f) = \mathcal{GP}(\mu, \Sigma)$, where the characteristics of f, such as smoothness and output magnitude, are fully defined by the kernel k (and its associated hyperparameters $\boldsymbol{\theta}$, which are omitted for brevity). We seek to inject an additional, user-defined prior belief over f into the GP, such as the prior over the optimum in Sec. 2.5, $\pi(\boldsymbol{x}) = \mathbb{P}\left(\boldsymbol{x} = \arg\max_{\boldsymbol{x}'} f(\boldsymbol{x}')\right)$. By postulating that π is accurate, we wish to form a belief-weighted prior - a prior over *functions* where the distribution over the optimum is exactly $\pi(\boldsymbol{x})$. We start by considering the user belief $\pi : \mathcal{X} \to \mathbb{R}$ from Eq. (4.3), and extend the definition to involve the integration over f, similarly to the Thompson sampling

definition of [30]. Formally,

$$\pi(\boldsymbol{x}) = \mathbb{P}\left(\boldsymbol{x} = \arg\max_{\boldsymbol{x}'} f(\boldsymbol{x}')\right) = \int_f \pi(\delta_*(\boldsymbol{x}|f))p(f)df \tag{4.4}$$

where $\delta_*(\boldsymbol{x}|f) = 1, \text{if } \boldsymbol{x} = \arg\max_{\boldsymbol{x}' \in \mathcal{X}} f(\boldsymbol{x}')$, and zero otherwise. As such, $\delta_*(\boldsymbol{x}|f)$ maps a function $f_i \sim p(f)$ to its arg max, and evaluates whether this arg max is equal to $\boldsymbol{x}$.

However, a belief over the optimum, or any other property, of a function f is implicitly a belief over the function f itself. As such, a non-uniform $\pi(\boldsymbol{x})$ should reasonably induce a change in the prior $p(f)$ to reflect the non-uniform optimum. To this end, we introduce an augmented user belief over the optimum $\rho^*_{\boldsymbol{x}} \sim \mathcal{P}^*_{\boldsymbol{x}}$, where $\mathcal{P}^*_{\boldsymbol{x}}$ is the prior over possible user beliefs, and draws are random functions $\rho^*_{\boldsymbol{x}} : \mathcal{X} \to \mathbb{R}^+$ which themselves take a function f as input, and output a positive real number quantifying the likelihood of a sample f_i under $\pi(\boldsymbol{x})$. Formally, we define $\rho^*_{\boldsymbol{x}}$ as

$$\rho^*_{\boldsymbol{x}}(f) = \mathbb{P}\left(\boldsymbol{x} = \arg\max_{\boldsymbol{x}'} f(\boldsymbol{x}')\right) = \frac{1}{Z_{\rho^*_{\boldsymbol{x}}}} \int_{\mathcal{X}} \delta_*(\boldsymbol{x}|f)\pi(\boldsymbol{x})d\boldsymbol{x} \tag{4.5}$$

where the intractible normalizing constant $Z_{\rho^*_{\boldsymbol{x}}}$ arises from the fact that the integrated density $\pi(\boldsymbol{x})$ acts on a finite-dimensional *property* of f, and not f itself. Under $\rho^*_{\boldsymbol{x}}(f)$, functions whose arg max lies in a high-density region under π will be assigned a higher probability. Notably, the definition in 4.5 can extend to other properties of f as well: a user belief p_{f_*} over the optimal value f_* analogously yields a belief over functions $\rho^*_{f_{\boldsymbol{x}}}(f)$:

$$\rho^*_{f_{\boldsymbol{x}}}(f) = \mathbb{P}\left(\boldsymbol{x} = \max_{\boldsymbol{x}'} f(\boldsymbol{x}')\right) = \frac{1}{Z_{\rho^*_{f_{\boldsymbol{x}}}}} \int_{f_{\boldsymbol{x}}} \delta_*(\boldsymbol{x}|f)p_{f*}(f_{\boldsymbol{x}})df_{\boldsymbol{x}}. \tag{4.6}$$

Notably, we integrate over $f_{\boldsymbol{x}}$ (and not $y_{\boldsymbol{x}}$) to signify that the optimal function value does not involve observation noise [61, 62]. It is worthwhile to reflect on the meaning of $\rho(f)$, and how beliefs over function properties propagate to $p(f)$. Concretely, if the user belief $\rho^*_{f_{\boldsymbol{x}}}(f)$ asserts that the maximal value lies within $C_1 < \max f < C_2$, the resulting distribution over f should only contain functions whose max falls within this range. Using rejection sampling, functions which disobey this criterion are filtered out, which yields the posterior $p(f|\rho)$. Having defined and exemplified how user beliefs impact the prior over functions $p(f)$, the role of ρ as a likelihood should be apparent: given a prior over functions $p(f)$ and a user belief over functions $\rho(f)$ which places a probability on all possible draws f_i $p(f)$, we can form a belief-weighted prior $p(f|\rho) \propto p(f)\rho(f)$. Thus, we introduce the formal definition of a user belief over a function property:

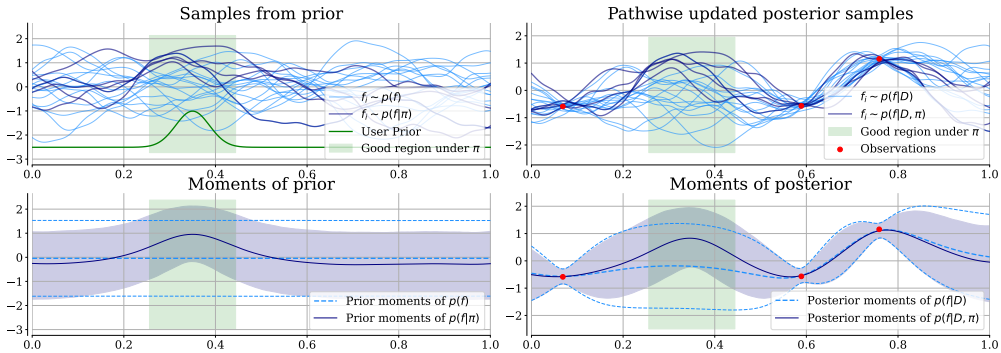

Fig. 4.2: (Top left) Draws from the prior $p(f)$ (light blue) and the belief-weighted prior $p(f|\rho)$ whose members are likely to have their optimum within the green region. (Top right) Pathwise updated draws based on observed data. As the green region is distant from the observed data, samples are almost unaffected by the data in this region. (Bottom left) Exact mean and standard deviation $(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}})$ of $p(f)$ and estimated mean and standard deviation of $p(f|\rho)$. (Bottom right) Exact $p(f|\mathcal{D})$ and estimated $p(f|\rho, \mathcal{D})$. As $p(f|\rho)$ constitutes of functions whose optimum is located within the green region the resulting model has a higher mean and lower variance within this region. Moreover, $p(f|\rho)$ globally displays lower upside variance compared to the vanilla GP.

Definition 1 (User Belief over Functions)**.** *The user belief over functions* $\rho(f) \propto \frac{p(f|\rho)}{p(f)}$.

As the subsequent derived methodology applies independently of the specific property of f that a prior is placed on, we will henceforth consider a belief over a general function property ρ. Having defined the role of ρ and the posterior over functions it produces, a natural question arises: *How is* $p(f|\rho)$ *updated once observations* $\mathcal{D}$ *are obtained?*

Since the data $\mathcal{D}$ is independent of the prior (the data generation process is intrinsically unaffected by the belief held by the user), application of Bayes' rule yields the following posterior $p(f|\mathcal{D}, \rho)$,

$$p(f|\mathcal{D}, \rho) = \frac{p(\mathcal{D}, \rho|f)p(f)}{p(\mathcal{D}, \rho)} \tag{4.7}$$

$$= \frac{p(\mathcal{D}|f)p(\rho|f)p(f)}{p(\mathcal{D})p(\rho)} \tag{4.8}$$

$$= \frac{p(f|\rho)}{p(f)} p(f|\mathcal{D}) \tag{4.9}$$

$$\propto \rho(f)p(f|\mathcal{D}), \tag{4.10}$$

where the right side of the proportionality in Eq. 4.7 suggests an intuitive

generation process for samples $(f|\mathcal{D},\rho)$ to approximate the density $p(f|\mathcal{D},\rho)$. Utilizing the pathwise update from Eq. 4.2, we note that given an approximate draw $\hat{f}$ from the prior, the subsequent data-dependent update is deterministic. Recalling Eq. 4.2 and assuming independence between ρ and $\mathcal{D}$, ρ only affects the draw from the prior, whereas $\mathcal{D}$ only affects the update. Consequently, we obtain

$$(\hat{f}|\mathcal{D},\rho)(\boldsymbol{x}) \overset{d}{=} \underbrace{(\hat{f}|\rho)(\boldsymbol{x})}_{\text{draw from prior}} + \underbrace{\mathbf{k}_n(\boldsymbol{x})^\top(\mathbf{K}_n+\sigma_\varepsilon^2\mathbf{I})^{-1}(\mathbf{y}-(\hat{f}|\rho)(\boldsymbol{x})-\boldsymbol{\varepsilon})}_{\text{deterministic update}}, \quad (4.11)$$

where $(\hat{f}|\rho) \sim p(f)\rho(\hat{f})$ are once again obtained using rejection sampling on draws from $p(\hat{f})$. Figure 4.2 displays this in detail: given the typical GP prior over functions *and* a user belief over the optimum, we obtain a distribution over functions $p(\hat{f}|\rho^*_{\boldsymbol{x}})$ before having observed any data (top right). Samples from the approximate prior $p(\hat{f})$ (light blue) are re-sampled proportionally to their probability of occurring under the prior $\rho^*_{\boldsymbol{x}}(\hat{f})$ in green, leaving samples $(\hat{f}|\rho^*_{\boldsymbol{x}})$ in navy blue, which are highly probable under $\rho^*_{\boldsymbol{x}}$. Once data is obtained, these samples are updates according to Eq. 4.11, which preserves the shape of the samples far away from observed data and yields the desired posterior.

3.2 Prior-weighted Monte Carlo Acquisition Functions

Naturally, neither the belief-weighted prior $p(f|\rho)$ nor the belief-weighted posterior $p(f|\mathcal{D},\rho)$ have a closed-form expression. Both are inherently non-Gaussian for non-uniform beliefs. As such, we resort to MC acquisition functions to compute utilities that are amenable to BO. In the subsequent section, we focus on the prevalent acquisition functions `EI`, and `MES`.

Expected Improvement The computation of the MC-`EI` within the `ColaBO` framework requires only minor adaptations of the original MC acquisition function. By definition, MC-`EI` assigns utility u as $u_{\texttt{EI}}(f(\boldsymbol{x})) = \max(f_n^* - f(\boldsymbol{x}), 0)$, which yields

$$\alpha_{\texttt{EI}}(\boldsymbol{x};\mathcal{D}) = \mathbb{E}_{f_{\boldsymbol{x}}|\mathcal{D}}[u_{\texttt{EI}}(f_{\boldsymbol{x}})] \approx \quad (4.12)$$

$$\sum_\ell \max(f_n^* - f_{\boldsymbol{x}}^{(\ell)}, 0),\ f_{\boldsymbol{x}}^{(\ell)} \sim p(f(\boldsymbol{x})|\mathcal{D}). \quad (4.13)$$

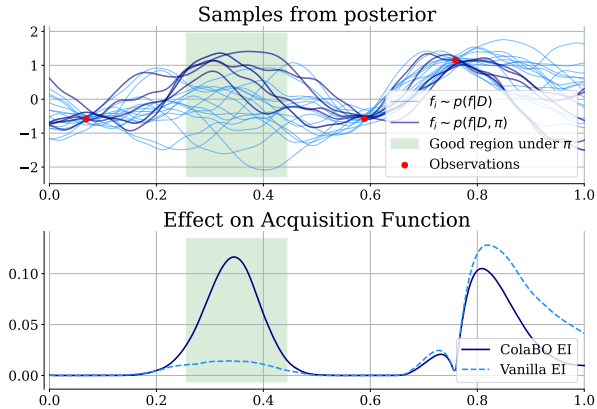

Fig. 4.3: (Top) Draws from $p(f|\mathcal{D})$ (light blue) and $p(f|\rho, \mathcal{D})$ with a prior ρ located in the green region. (Bottom) Vanilla MC-`EI` and `ColaBO` MC-`EI`, resulting from computing the acquisition function from sample draws from $p(f|\rho, \mathcal{D})$.

Utilizing rejection sampling, we can compute the MC-`EI` under the `ColaBO` posterior accordingly,

$$\alpha_{\texttt{EI}}(\boldsymbol{x}; \mathcal{D}, \rho) = \mathbb{E}_{f_{\boldsymbol{x}}|\mathcal{D},\rho}[u_{\texttt{EI}}(f_{\boldsymbol{x}})] \tag{4.14}$$

$$\propto \int_f u_{\texttt{EI}}(f_{\boldsymbol{x}})\rho(f)p(f|\mathcal{D})df \tag{4.15}$$

$$\approx \sum_{\ell} \rho(f^{(\ell)}) \max(f_n^* - f_{\boldsymbol{x}}^{(\ell)}, 0), \tag{4.16}$$

where

$$f_{\boldsymbol{x}}^{(\ell)} \sim p(f(\boldsymbol{x})|\mathcal{D}), \tag{4.17}$$

and wherein samples in Eq. 4.16 are drawn from the prior, retained with probability $\rho(f^{(\ell)})/\max \rho$, and pathwise updated. In Figure 4.3, we demonstrate how `ColaBO-EI` differs from MC-`EI` for an identical posterior as in Figure 4.2. By computing $\alpha_{\texttt{EI}}$ from samples biased by ρ, `ColaBO` substantially directs the search towards good regions under ρ. Derivations for `PI` and `KG` are analogous to that of `EI`.

Max-Value Entropy Search We derive a `ColaBO-MES` acquisition function by first considering the definition of the entropy, $\mathrm{H}[p(y_{\boldsymbol{x}}|\mathcal{D})] = \mathbb{E}_{y_{\boldsymbol{x}}|\mathcal{D}}[-\log p(y_{\boldsymbol{x}}|\mathcal{D})]$. When considering the belief-weighted posterior, we further condition the posterior

Algorithm 4.1 `ColaBO` iteration

Input: User prior ρ, number of function samples L, current data $\mathcal{D}$
Output: Next query location $\boldsymbol{x}'$.
for $\ell \in \{1, \dots, L\}$ **do**
  $\rho^{(\ell)} = \rho(\hat{f}^{(\ell)}; n), \hat{f}^{(\ell)} \sim p(\hat{f})$ {`Sample functions and evaluate on` π}
  $(\hat{f}^{(\ell)}|\mathcal{D}) =$ `PathwiseUpdate`$(\hat{f}^{(\ell)}, \mathcal{D})$ {`Per-sample update as in Eq. 4.11`}
end for
$p(\hat{f}|\mathcal{D}, \rho) \approx \sum_\ell \rho^{(\ell)}(\hat{f}^{(\ell)}|\mathcal{D})$ {`Form MC estimate of posterior`}
$\boldsymbol{x}' = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}_{p(\hat{f}|\mathcal{D},\rho)}[u(\hat{f}_{\boldsymbol{x}})]$ {`Maximize MC acquisition`}

on ρ and obtain

$$\alpha_{\texttt{MES}}(\boldsymbol{x}) = \mathbb{E}_{f^*|\mathcal{D},\rho}\left[\mathbb{E}_{y_{\boldsymbol{x}}|\mathcal{D},\rho,f^*}[\log p(y_{\boldsymbol{x}}|\mathcal{D}, \rho, f^*)]\right] \tag{4.18}$$

$$- \mathbb{E}_{y_{\boldsymbol{x}}|\mathcal{D},\rho}[\log p(y_{\boldsymbol{x}}|\mathcal{D}, \rho)] \tag{4.19}$$

$$\propto \mathbb{E}_{f^*|\mathcal{D},\rho}\left[\mathbb{E}_{f_{\boldsymbol{x}}|\mathcal{D},\rho}[\mathbb{E}_{y_{\boldsymbol{x}}|f_{\boldsymbol{x}}}[\log p(y_{\boldsymbol{x}}|f_{\boldsymbol{x}}, \rho, f^*)]]\right] \tag{4.20}$$

$$- \mathbb{E}_{f_{\boldsymbol{x}}|\mathcal{D},\rho}[\mathbb{E}_{y_{\boldsymbol{x}}|f_{\boldsymbol{x}}}[\log p(y_{\boldsymbol{x}}|f_{\boldsymbol{x}}, \rho)]] \tag{4.21}$$

$$\approx \frac{1}{Z_J} \sum_{j=1}^{J} \sum_{\ell=1}^{L} \sum_{k=1}^{K} \log p(y_{\boldsymbol{x}}^{(k)}|f_{\boldsymbol{x}}^{(\ell)}, f_*^{(j)}) \rho(f^{(\ell)}) \rho(f^{(j)}) \tag{4.22}$$

$$- \sum_{\ell=1}^{L} \sum_{k=1}^{K} \log p(y_{\boldsymbol{x}}^{(k)}|f_{\boldsymbol{x}}^{(\ell)}) \rho(f^{(\ell)}), \tag{4.23}$$

where Z_J is a normalizing constant $\sum_J \rho(f^{(j)})$ brought on by sampling optimal values, $y_{\boldsymbol{x}}|f_{\boldsymbol{x}}$ can trivially be obtained by sampling Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ to a noiseless observation $f_{\boldsymbol{x}}|\mathcal{D}$ in the innermost expectation, and $f_{\boldsymbol{x}}$ and f^* are obtained through the pathwise sampling procedure outlined in Eq. 4.11. The samples are evaluated on $p((y_{\boldsymbol{x}}|f_{\boldsymbol{x}}),(y_{\boldsymbol{x}}|f_{\boldsymbol{x}}, f^*))$. As evident by Eq. 4.23, ρ affects the posterior distribution of both the observations $y_{\boldsymbol{x}}$ and the optimal values f^*. `PES` and `JES` are derived analogously. However, these acquisition function require conditioning on additional, simulated data and consequently, additional pathwise updates, to compute.

3.3 Practical Considerations

`ColaBO` introduces additional flexibility to MC-based BO acquisition functions. The `ColaBO` framework deviates from vanilla (q-)MC acquisition functions [72, 3] by utilizing approximate sample functions from the posterior, as opposed to pointwise draws from the posterior predictive and the reparametrization trick [51].

ColaBO holds three shortcomings not prevalent in vanilla MC acquisition functions: (1) it cannot utilize Quasi-MC in the draws from the predictive posterior (only in the RFF weights), (2) it cannot fix the base samples [3] drawn from the posterior for acquisition function consistency across the search space, and (3) the RFF approximation of $p(f)$ introduces bias. This approximation error is more pronounced for the Matérn 5/2-kernel than the squared exponential, leaving ColaBO best suited for the latter. In Sec. 4.1, we display the impact of these shortcomings. While acquisition function optimization no longer enjoys the improved accuracy that stems from the reparametrization trick, the high degree of smoothness of function samples still allow for efficient gradient-based optimization.

4 Results

We evaluate the performance of ColaBO on various tasks, using priors over the optimum $\rho_{\boldsymbol{x}^*}$ obtained from known optima on synthetic tasks, as well as from prior work [37] on realistic tasks. We consider two variants of ColaBO: one using LogEI [1], a numerically stable, smoothed logsumexp transformation of EI with analogous derivation, and one variant using MES. We benchmark against the vanilla variants of each acquisition function, as well as πBO [24] and decoupled Thompson sampling [63, 73]. All acquisition functions are implemented in BoTorch [3] using a squared exponential kernel and MAP hyperparameter estimation. Unless stated otherwise, all methods are initialized with the mode of the prior followed by 2 Sobol samples. Our code is publicly available at https://github.com/hvarfner/colabo.

4.1 Approximation Quality of the ColaBO Framework

Firstly, we demonstrate the approximation quality of ColaBO *without* user priors to assert its accuracy compared to a vanilla MC acquisition function. To facilitate comparison, we randomly sample 10 points on the Hartmann (3D) function, and optimize LogEI with a large budget. We subsequently optimize ColaBO-LogEI on the same set of points and compare the arg max to the solution found by the gold standard. Figure 4.4 displays the (log10) Euclidian distance between the arg max of LogEI and its ColaBO variant. We note that, for small amounts ($\leqslant 256$) of posterior samples, the error induced by RFF bias is relatively low,

which is evidenced by all RFF variants being roughly equal in distance to the true acquisition function optimizer.

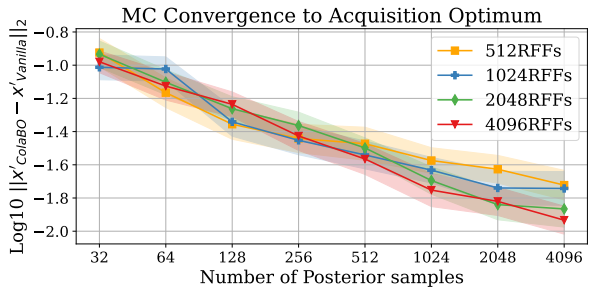

Fig. 4.4: Mean and $1/4$ standard deviation of MC-induced errors of `ColaBO-LogEI` relative vanilla LogEI as measured by the distance to the $\arg\max$ of the acquisition function on Hartmann (3D) on 10 randomly sampled points for 40 seeds.

4.2 Synthetic Functions with Known Priors

We adapt a similar evaluation protocol to [24], and evaluate `ColaBO` for two types of user beliefs for synthetic tasks: well-located and poorly located priors over the optimal location, designed to emulate a well-informed and poorly-informed practitioner, respectively. The well-located prior is offset by a small (10%) amount from the optimum, and the poorly located prior is maximally offset, while retaining its mode inside the search space. On well-located priors, both `ColaBO-LogEI` and `ColaBO-MES` demonstrate substantially improved performance relative to their vanilla counterparts, comparable to π`BO` on all benchmarks. On poorly located priors, `ColaBO` demonstrates superior robustness, recovering the performance of the vanilla acquisition function within the maximal budget of $20D$ iterations and clearly outperforming π`BO`, which more frequently misled by the poor prior.

4.3 Hyperparameter Tuning tasks

For the real-world HPO tasks, we consider two different benchmarking suites: LCBench [76] and PD1 [69]. For LCBench, we evaluate all methods on five deep learning tasks (6D). While the optima for these tasks are ultimately unknown, we utilize the priors provided in MF-Prior-Bench [5] [37], which are intended to provide a good starting point for further optimization. The chosen tasks

[5] `https://github.com/automl/mf-prior-bench`

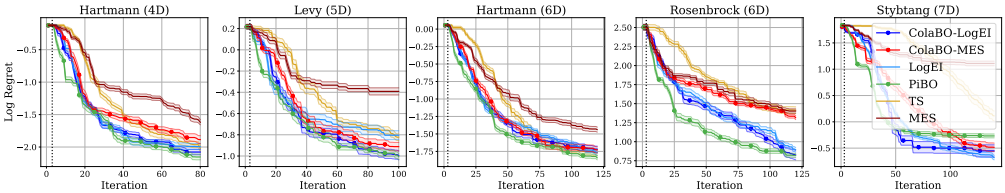

Fig. 4.5: Performance on synthetic functions with well-located priors. Both `ColaBO-LogEI` and `ColaBO-MES` offer drastic speed-ups over their vanilla variants, and offer similar performance to π`BO`. The ranking of `ColaBO` acquisition functions are generally consistent with their respective vanilla variants. This is most prominent on Rosenbrock (6D), where `ColaBO-MES` struggles similarly to vanilla `MES`.

were the five tasks with available priors of the best (good) strength, as per the benchmark suite. To emulate a realistic HPO setting, we consider a smaller optimization budget of 40 iterations, and initialize all methods that utilize user beliefs with only one initial sample, that being the mode of the prior. Figure 4.7 shows the performance of all methods on the LCBench tasks. `ColaBO` improves substantially on the baseline approaches for 3 out of 5 tasks. π`BO` is the overall best-performing method, followed by `ColaBO-LogEI`.

Lastly, we evaluate `ColaBO` on three $4D$ deep learning HPO tasks from the PD1 [69] benchmarking suite, once again using priors from MF-Prior-Bench. The two `ColaBO` variants perform best in this evaluation, producing the best terminal performance on two tasks (CIFAR, LM1B), with all methods being tied on the third (CIFAR). `ColaBO` demonstrates consistent speed-ups compared to its vanilla counterparts, surpassing the terminal performance of the baseline within a third of the budget on CIFAR and LM1B.

5 Related Work

In BO, auxiliary prior information can be conveyed in multiple ways. We outline meta learning/transfer learning for BO based on data from previous experiments, and data-less approaches.

Learning from Previous Experiments Transfer learning and meta learning for BO aims to automatically extract and use knowledge from prior executions of BO by pre-training the model on data acquired from previous executions [60, 74, 48, 11, 12, 52, 53, 75, 13]. Typically, meta- and transfer learning exploit relevant previous data for training the GP for the current task while retaining predictive

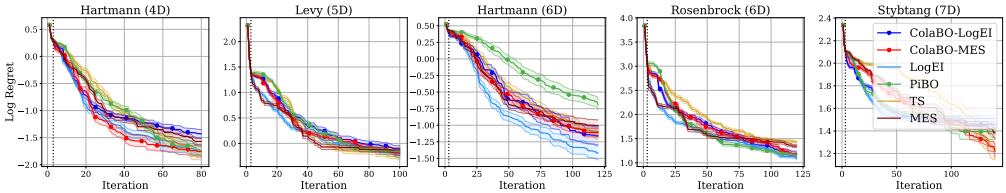

Fig. 4.6: Performance on poorly located priors. `ColaBO` acquisition functions are more robust than π`BO`, as it frequently recovers the performance of the vanilla acquisition function before the total budget is depleted. `ColaBO-LogEI` struggles marginally on Hartmann (6D). `ColaBO-MES` recovers the baseline on all tasks.

uncertainty to account for imperfect task correlation.

Expert Priors over Function Optimum Few previous works have proposed to inject explicit prior distributions over the location of an optimum into BO. In these cases, users explicitly define a prior that encodes their beliefs on where the optimum is more likely to be located. [4] suggest an approach that supports prior beliefs from a fixed set of distributions, which affects the very initial stage of optimization. However, this approach cannot be combined with standard acquisition functions. BOPrO [58] employs a similar structure that combines the user-provided prior distribution with a data-driven model into a pseudo-posterior. From the pseudo-posterior, configurations are selected using the EI acquisition function, using the formulation in [4]. πBO [24] suggests a general-purpose prior-weighted acquisition function, where the influence of the prior decreases over time. They provide convergence guarantees for when the framework is applied to the EI acquisition function. While effective, none of these approaches act on the surrogate model in a Bayesian-principled fashion, but strictly as heuristics. Moreover, they solely focus on priors over optimal inputs, thus offering less utility than `ColaBO`.

Priors over Optimal Value Similarly few works have addressed the issue of auxilliary knowledge of the optimal value. Both [26] and [46] propose altering the GP and accompanying it with tailored acquisition functions. [26] employ variational inference, proposing distinct variational families depending on the type of knowledge pertaining to the optimal value. [46] use a parabolic transformation of the output space to ensure an upper bound is preserved. Unlike `ColaBO`, neither of these methods is general enough to accompany arbitrary user priors to guide the optimization.

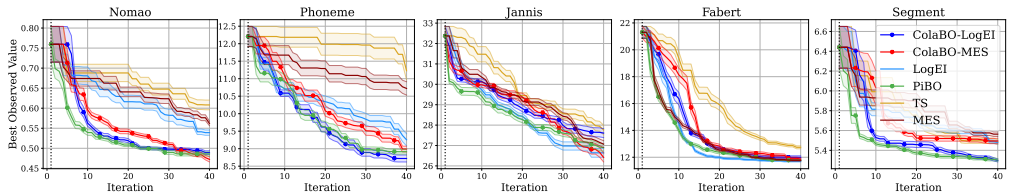

Fig. 4.7: Performance on the 6D LCBench hyperparameter tuning tasks of various deep learning pipelines. `ColaBO` substantially improves on the non-prior baselines for 3 out of five tasks. π`BO` performs best on aggregate, and achieves the best acceleration in performance at early iterations.

6 Conclusion, Limitations and Future Work

We presented `ColaBO`, a flexible BO framework that allows practitioners to inject beliefs over function properties in a Bayesian-principled manner, allowing for increased efficiency in the BO procedure. `ColaBO` works across a collection of MC acquisition functions, inheriting their flexibility in batch optimization and ability to work with non-Gaussian posteriors. It demonstrates competitive performance for well-located priors, using them to substantially accelerate optimization. Moreover, it retains approximately baseline performance when applied to detrimental priors, demonstrating greater robustness than π`BO`. `ColaBO` crucially relies on multiple steps of MC. While flexible, this approach drives computational expense in order to assert sufficient accuracy, requiring tens of seconds per evaluation to achieve desired accuracy, depending on the size of the benchmark. Moreover, obtaining draws from $\rho^*_{\boldsymbol{x}}$ scales exponentially in the dimensionality of the prior. While practitioners are unlikely to specify priors over more than a handful of variables, `ColaBO` may become impractical when priors of higher dimensionality are employed. Paths for future work could involve more accurate and efficient sampling procedures [35] from the belief-weighted prior, as well as variational [64] or pre-trained [41, 42] approaches to obtain a representative belief-biased model with an analytical posterior. This would likely bring down the runtime of `ColaBO` and broaden its potential use. Lastly, applying `ColaBO` to multi-fidelity optimization [29, 37] offers an additional avenue for increased efficiency which would further increase its viability on costly deep learning pipelines.

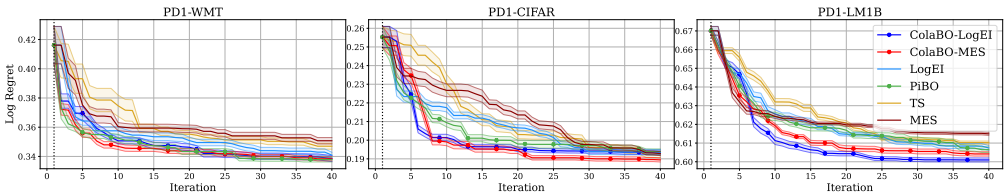

Fig. 4.8: Performance on the 4D PD1 hyperparameter tuning tasks of various deep learning pipelines. `ColaBO` drastically accelerates optimization initially, finding configurations with close to terminal performance quickly. π`BO` offers competitive performance, but lacks the rapid initial progress of `ColaBO` on CIFAR and LM1B.

References

[1] Sebastian Ament, Sam Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for bayesian optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=1vyAG6j9PE`.

[2] Raul Astudillo and Peter Frazier. Bayesian optimization of function networks. *Advances in neural information processing systems*, 34:14463–14475, 2021.

[3] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. In *Advances in Neural Information Processing Systems*, 2020. URL `http://arxiv.org/abs/1910.06403`.

[4] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Proceedings of the 25th International Conference on Advances in Neural Information Processing Systems (NeurIPS'11)*, pages 2546–2554, 2011.

[5] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24. Curran Associates, Inc., 2011.

[6] E. Brochu, V. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv:1012.2599v1 [cs.LG]*, 2010.

[7] Adam D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12:2879–2904, 2011.

[8] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. Deisenroth. Bayesian gait optimization for bipedal locomotion. In P. Pardalos and M. Resende, editors, *Proceedings of the Eighth International Conference on Learning and Intelligent Optimization (LION'14)*, Lecture Notes in Computer Science. Springer, 2014.

[9] Adel Ejjeh, Leon Medvinsky, Aaron Councilman, Hemang Nehra, Suraj Sharma, Vikram Adve, Luigi Nardi, Eriko Nurvitadhi, and Rob A Rutenbar. HPVM2FPGA: Enabling True Hardware-Agnostic FPGA Programming. In *Proceedings of the 33rd IEEE International Conference on Application-specific Systems, Architectures, and Processors*, 2022.

[10] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

[11] M. Feurer, Jost Tobias Springenberg, and F. Hutter. Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1128–1135, 2015.

[12] M. Feurer, B. Letham, F. Hutter, and E. Bakshy. Practical transfer learning for bayesian optimization. *ArXiv abs/1802.02219*, 2018.

[13] Matthias Feurer, Benjamin Letham, Frank Hutter, and Eytan Bakshy. Practical transfer learning for Bayesian optimization. *arXiv preprint 1802.02219*, 2022.

[14] P. Frazier, W. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM J. Control and Optimization*, 47: 2410–2439, 01 2008. doi: 10.1137/070693424.

[15] Peter I Frazier and Jialei Wang. Bayesian optimization for materials design. In *Information science for materials discovery and design*, pages 45–75. Springer, 2016.

[16] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023. to appear.

[17] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 2020.

[18] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(1):1809–1837, June 2012. ISSN 1532-4435.

[19] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, 2014. URL `https://proceedings.neurips.cc/paper/2014/file/069d3bb002acd8d7dd095917f9efe4cb-Paper.pdf`.

[20] José Miguel Hernández-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In *International conference on machine learning*, pages 1699–1707. PMLR, 2015.

[21] Daolang Huang, Louis Filstroff, Petrus Mikkola, Runkai Zheng, and Samuel Kaski. Bayesian optimization augmented with actively elicited expert knowledge, 2022.

[22] F. Hutter, H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In C. Coello, editor, *Proceedings of the Fifth International Conference on Learning and Intelligent Optimization (LION'11)*, volume 6683 of *Lecture Notes in Computer Science*, pages 507–523. Springer, 2011.

[23] Carl Hvarfner, Frank Hutter, and Luigi Nardi. Joint entropy eearch for maximally-informed bayesian optimization. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022.

[24] Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. PiBO: Augmenting Acquisition Functions with User Beliefs for Bayesian Optimization. In *International Conference on Learning Representations*, 2022.

[25] Carl Hvarfner, Erik Hellsten, Frank Hutter, and Luigi Nardi. Self-correcting bayesian optimization through bayesian active learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=dX9MjUtP1A`.

[26] Taewon Jeong and Heeyoung Kim. Objective bound conditional gaussian process for bayesian optimization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4819–4828.

PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/jeong21a.html.

[27] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 12 1998. doi: 10.1023/A:1008306431147.

[28] A G Journel and C J Huijbregts. Mining geostatistics, Jan 1976.

[29] K. Kandasamy, G. Dasarathy, J. Oliva, J. Schneider, and B. Póczos. Gaussian Process Bandit Optimisation with Multi-fidelity Evaluations. In D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Proceedings of the 30th International Conference on Advances in Neural Information Processing Systems (NeurIPS'16)*, pages 992–1000, 2016.

[30] K. Kandasamy, A. Krishnamurthy, J. Schneider, and B. Póczos. Parallelised Bayesian optimisation via Thompson sampling. In A. Storkey and F Perez-Cruz, editors, *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84, pages 133–142. Proceedings of Machine Learning Research, 2018.

[31] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. URL https://arxiv.org/abs/1312.6114.

[32] Arun Kumar, Santu Rana, Alistair Shilton, and Svetha Venkatesh. Human-ai collaborative bayesian optimisation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 16233–16245. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6751611b394a3464cea53eed91cf163c-Paper-Conference.pdf.

[33] H. J. Kushner. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, 86(1):97–106, 03 1964. ISSN 0021-9223. doi: 10.1115/1.3653121. URL https://doi.org/10.1115/1.3653121.

[34] B. Letham, K. Brian, G. Ottoni, and E. Bakshy. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 2018.

[35] Jihao Andreas Lin, Javier Antorán, Shreyas Padhy, David Janz, José Miguel Hernández-Lobato, and Alexander Terenin. Sampling from gaussian process posteriors using stochastic gradient descent. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Sf9goJtTCE.

[36] Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. Smac3: A versatile bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23 (54):1–9, 2022. URL `http://jmlr.org/papers/v23/21-0888.html`.

[37] Neeratyoy Mallik, Edward Bergman, Carl Hvarfner, Danny Stoll, Maciej Janowski, Marius Lindauer, Luigi Nardi, and Frank Hutter. Priorband: Practical hyperparameter optimization in the age of deep learning. *arXiv preprint 2306.12370*, 2023.

[38] Matthias Mayr, Carl Hvarfner, Konstantinos Chatzilygeroudis, Luigi Nardi, and Volker Krueger. Learning skill-based industrial robot tasks with user priors. *IEEE 18th International Conference on Automation Science and Engineering*, 2022. URL `https://arxiv.org/abs/2208.01605`.

[39] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

[40] Henry B. Moss, David S. Leslie, Javier Gonzalez, and Paul Rayson. Gibbon: General-purpose information-based bayesian optimisation. *Journal of Machine Learning Research*, 22(235):1–49, 2021. URL `http://jmlr.org/papers/v22/21-0120.html`.

[41] Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=KSugKcbNf9`.

[42] Samuel Müller, Matthias Feurer, Noah Hollmann, and Frank Hutter. PFNs4BO: In-context learning for Bayesian optimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 25444–25470. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/muller23a.html`.

[43] Mojmir Mutny and Andreas Krause. Efficient high dimensional bayesian optimization with additivity and quadrature fourier features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/4e5046fc8d6a97d18a5f54beaed54dea-Paper.pdf`.

[44] L. Nardi, D. Koeplinger, and K. Olukotun. Practical design space exploration. In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 347–358. IEEE, 2019.

[45] Willie Neiswanger, Ke Alexander Wang, and Stefano Ermon. Bayesian algorithm execution: Estimating computable properties of black-box functions using mutual information. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8005–8015. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/neiswanger21a.html`.

[46] Vu Nguyen and Michael A. Osborne. Knowing the what but not the where in Bayesian optimization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7317–7326. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/nguyen20d.html`.

[47] C. Oh, E. Gavves, and M. Welling. BOCK : Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*, pages 3865–3874, 2018.

[48] V. Perrone, H. Shen, M. Seeger, C. Archambeau, and R. Jenatton. Learning search spaces for bayesian optimization: Another view of hyperparameter transfer learning. In *Advances in Neural Information Processing Systems*, 2019.

[49] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL `https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf`.

[50] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[51] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. URL `https://proceedings.mlr.press/v32/rezende14.html`.

[52] Jonas Rothfuss, Vincent Fortuin, Martin Josifoski, and Andreas Krause. Pacoh: Bayes-optimal meta-learning with pac-guarantees. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9116–9126, 2021.

[53] Jonas Rothfuss, Dominique Heyn, Jinfan Chen, and Andreas Krause. Meta-learning reliable priors in the function space. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[54] Binxin Ru, Xingchen Wan, Xiaowen Dong, and Michael Osborne. Interpretable neural architecture search via bayesian optimisation with weisfeiler-lehman kernels. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=j9Rv7qdXjd`.

[55] B. Shahriari, K. Swersky, Z. Wang, R. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[56] L. Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[57] J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian optimization of machine learning algorithms. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems (NeurIPS'12)*, pages 2960–2968, 2012.

[58] A. Souza, L. Nardi, L. Oliveira, K. Olukotun, M. Lindauer, and F. Hutter. Bayesian optimization with a prior for the optimum. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part III*, volume 12977 of *Lecture Notes in Computer Science*, pages 265–296. Springer, 2021.

[59] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, May 2012. ISSN 1557-9654. doi: 10.1109/tit.2011.2182033. URL `http://dx.doi.org/10.1109/TIT.2011.2182033`.

[60] K. Swersky, J. Snoek, and R. Adams. Multi-task Bayesian optimization. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Proceedings of the 27th International Conference on Advances in*

Neural Information Processing Systems (NeurIPS'13), pages 2004–2012, 2013.

[61] Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9334–9345. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/takeno20a.html`.

[62] Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Sequential and parallel constrained max-value entropy search via information lower bound. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20960–20986. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/takeno22a.html`.

[63] W. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[64] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In David van Dyk and Max Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL `https://proceedings.mlr.press/v5/titsias09a.html`.

[65] Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective bayesian optimization. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=ZChgD8OoGds`.

[66] Q. Wang, Y. Ming, Z. Jin, Q. Shen, D. Liu, M. J. Smith, K. Veeramachaneni, and H. Qu. Atmseer: Increasing transparency and controllability in automated machine learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12. Association for Computing Machinery, 2019.

[67] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning (ICML)*, 2017.

[68] Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 745–754. PMLR, 09–11 Apr 2018. URL `https://proceedings.mlr.press/v84/wang18c.html`.

[69] Zi Wang, George E. Dahl, Kevin Swersky, Chansoo Lee, Zachary Nado, Justin Gilmer, Jasper Snoek, and Zoubin Ghahramani. Pre-trained Gaussian processes for Bayesian optimization. *arXiv preprint arXiv:2109.08215*, 2023.

[70] C. White, W. Neiswanger, and Y. Savani. BANANAS: Bayesian optimization with neural architectures for neural architecture search. In *Association for the Advancement of Artificial Intelligence*, pages 10293–10301, 2021.

[71] James Wilson, Frank Hutter, and Marc Deisenroth. Maximizing acquisition functions for bayesian optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper/2018/file/498f2c21688f6451d9f5fd09d53edda7-Paper.pdf`.

[72] James T. Wilson, Riccardo Moriconi, Frank Hutter, and Marc Peter Deisenroth. The reparameterization trick for acquisition functions, 2017. URL `https://arxiv.org/abs/1712.00424`.

[73] James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Efficiently sampling functions from gaussian process posteriors. In *International Conference on Machine Learning*, 2020. URL `https://arxiv.org/abs/2002.09309`.

[74] M. Wistuba, N. Schilling, and L. Schmidt-Thieme. Hyperparameter search space pruning - A new component for sequential model-based hyperparameter optimization. In A. Appice, P. Rodrigues, V. Costa, J. Gama, A. Jorge, and C. Soares, editors, *Machine Learning and Knowledge Discovery in Databases (ECML/PKDD'15)*, volume 9285 of *Lecture Notes in Computer Science*, pages 104–119. Springer, 2015.

[75] Martin Wistuba and Josif Grabocka. Few-shot bayesian optimization with deep kernel surrogates. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=bJxgv5C3sYc`.

[76] Lucas Zimmer, Marius Lindauer, and Frank Hutter. Auto-pytorch tabular: Multi-fidelity metalearning for efficient and robust autodl. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):3079 – 3090, 2021.

Paper V

Leveraging Axis-Aligned Subspaces for High-dimensional Bayesian Optimization with Group Testing

Erik Hellsten* Lund University

Carl Hvarfner* Lund University

Leonard Papenmeier* Lund University

Luigi Nardi Lund University

Abstract

Bayesian optimization (BO) is an effective method for optimizing expensive-to-evaluate black-box functions, but its susceptibility to the curse of dimensionality limits its applicability to high-dimensional problems. The assumption of an axis-aligned active subspace, where few dimensions have a significant impact on the objective, motivated several algorithms for high-dimensional BO. However, the validity of this assumption is rarely verified, and the assumption is rarely exploited to its full extent. We propose a group testing (GT) approach to identify active variables to facilitate efficient optimization in these domains. The proposed algorithm, Group Testing Bayesian Optimization (`GTBO`), first runs a testing phase where groups of variables are systematically selected and tested on whether they influence the objective. To that end, we extend the well-established GT theory to functions over continuous domains. In the second phase, `GTBO` guides optimization by placing more importance on the active dimensions. By leveraging the axis-aligned subspace assumption, `GTBO` is competitive against state-of-the-art methods on benchmarks satisfying the assumption of axis-aligned subspaces. Furthermore, for a given application, `GTBO` helps discover the active variables, enhancing practitioners' understanding and explainability of the problem.

1 Introduction

Noisy and expensive-to-evaluate black-box functions occur in many practical optimization tasks, including material design [53], hardware design [33, 18], hyperparameter tuning [26, 41, 11], and robotics [10, 6, 32]. BO is an established framework that allows optimization of such problems in a sample-efficient manner [45, 21]. Despite its many advantages, BO faces challenges with the curse of dimensionality, limiting its effectiveness in high-dimensional domains like robotics [10], joint neural architecture search and hyperparameter optimization [5], drug discovery [35], chemical engineering [9], and vehicle design [25].

In recent years, efficient approaches have been proposed to tackle the limitations of BO in high dimensions. Many of these approaches assume the existence of a low-dimensional *active subspace* of the input domain that has a significantly larger impact on the optimization objective than its complement [50, 28]. Often, the active subspace is further assumed to be axis-aligned [34, 19, 47, 37, 38], i.e., only a set of all considered variables impact the objective. The validity of this simplifying assumption does not always hold in real-world applications, and as a consequence, several approaches relax this assumption to reduce the risk of failure [19, 37, 38]. Those relaxations, however, come at a cost: firstly, they negatively affect sample efficiency for problems with axis-aligned subspaces, and secondly, they dilute the insights into which variables are relevant to the application at hand. Instead, we aim to leverage those assumptions more strongly, yielding better performance and stronger insights when they hold.

Knowing the active dimensions of a problem yields additional insight into the application, informing the user which problem parameters deserve more attention. When the active subspace is axis-aligned, finding the active dimensions can be framed as a feature selection problem. A straightforward approach is first to learn the active dimensions using a dedicated feature selection approach and subsequently optimize over the learned subspace. We propose to initially find the active dimensions using an information-theoretic approach built around the well-established theory of group testing [17]. Group testing is the problem of finding several active elements within a larger set by iteratively testing groups of elements. We develop the theory needed to transition noisy GT, which otherwise only allows binary observations, to support evaluations of continuous black-box functions. This enables GT in BO and other applications, such as feature selection for regression problems. The contributions of this work are:

1. We extend the theory of group testing to feature importance analysis in a continuous setting tailored towards Gaussian process (GP) modeling.

2. We introduce Group Testing Bayesian Optimization (GTBO), a BO method that, based on the assumption of axis-aligned active subspaces, leverages the activeness information obtained from the preceding GT phase to guide the optimization.
3. We demonstrate that GTBO is competitive against state-of-the-art high-dimensional methods and reliably identifies active dimensions with high probability when the underlying assumptions hold.

2 Background

2.1 High-dimensional Bayesian optimization

We aim to find a minimizer $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$ of the black-box function $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$, over the D-dimensional input space $\mathcal{X} = [0,1]^D$. We assume that f can only be observed point-wise and that the observation is perturbed by noise, $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$, where σ_n^2 is the *noise variance*. We further assume f to be expensive to evaluate, so the number of function evaluations is limited. In this work, we consider problems of high dimensionality D, where only d_e dimensions are *active*, and the other $D - d_e$ dimensions are *inactive*. Here, inactive means that the function value changes only marginally along the inactive dimensions compared to the active dimensions to the extent that satisfactory optimization performance can be obtained by considering the active dimensions alone. We refer the reader to [21] for an in-depth introduction to BO.

A popular remedy to the curse of dimensionality is trust regions [39, 40], where, instead of reducing the dimensionality, one optimizes over a hyper-rectangle in input space. This makes the algorithm more local to counteract the over-exploration exhibited by traditional BO in high dimensions. One successful approach in this category is TuRBO [19]. Even though TuRBO operates in the full input dimensionality and might not scale to arbitrarily high-dimensional problems, it has shown remarkable performance in several applications. CASMOPOLITAN [49] extends TuRBO to mixed and combinatorial spaces.

2.2 Low-dimensional subspace Bayesian optimization

Using linear embeddings is a common approach when optimizing high-dimensional functions that contain a low-dimensional active subspace. REMBO [50] shows that

a random embedded subspace with at least the same dimensionality as the active subspace is guaranteed to contain an optimum if the subspace is unbounded. However, BO usually requires a bounded search space, and `REMBO` suffers from projecting outside of this search space. `HeSBO` [34] uses a sparse projection matrix to avoid points outside the search space. `BAxUS` [37] and `Bounce` [38] use an `HeSBO`-like embedding [34] but allow the dimensionality of the target space to grow over time. This ensures that the optimum can eventually found but leads to `BAxUS` and `Bounce` optimizing over high-dimensional spaces in later optimization stages. `Alebo` [27] presents another remedy to shortcomings in the search space design of `REMBO`. In particular, bounds from the original space are projected into the embedded space, and the kernel in the embedded space is adjusted to preserve distances from the original space. Notably, methods that rely on random embeddings require the user to provide a guess on the effective dimensionality of the problem, which might be challenging for real-world applications.

Axis-aligned active subspaces. One common assumption to tackle high-dimensional problems is that the active subspace is axis-aligned, i.e., a subspace that can be obtained by removing the inactive dimensions. This is equivalent to the assumption of active and inactive dimensions. `SAASBO` [19] sets up on this assumption by adding a strong sparsity-inducing prior to the hyperparameters of the GP model, prioritizing fewer active dimensions unless the data strongly suggests otherwise. `VS-BO` [46] actively identifies relevant variables in a problem, similar to our approach. However, it relies on a heuristic tied to a specific surrogate model (GP). It lacks a clear-cut decision on variable relevance due to ongoing variable importance estimation during optimization. `MCTS-VS` [47] uses a Monte Carlo tree search to select the active dimensions dynamically. It relies on randomly chosen sets of active and inactive dimensions, which can make finding the "correct" active dimensions difficult if the number of active dimensions is high. Our work also leverages the axis-aligned assumption, but it differs in the way we identify the active dimensions.

Active subspace learning. In this paper, we resolve to a more direct approach, where we learn the active subspace explicitly. This is frequently denoted by *active subspace learning*. A common approach is to divide the optimization into two phases. The first phase involves selecting points and analyzing the structure to find the subspace. An optimization phase then follows on the subspace that was identified. The initial phase can also be used alone to gain insights into the problem. One of the more straightforward approaches is to look for linear trends using methods such as *principal component analysis* [48] or *partial least*

squares [8]. [16] use low-rank matrix recovery with directional derivatives with finite differences to find the active subspace. If gradients are available, the active subspace is spanned by the eigenvectors of the matrix $C := \int_{\mathcal{X}} \nabla f(x)(\nabla f(x))^T dx$ with non-zero eigenvalues. This is used by [13] and [52] to show that C can be estimated in closed form for GP regression. We refer to the survey by [7] for a more in-depth introduction to active subspace learning. Notably, large parts of the active subspace learning literature yield non-axis-aligned subspaces. This is disadvantageous for problems with axis-aligned subspaces, as the information about the active dimensions is diluted.

2.3 Group testing

Group testing (GT, [1]) is a methodology for identifying elements with some low-probability characteristic of interest by jointly evaluating groups of elements. GT was initially developed to test for infectious diseases in larger populations but has later been applied in quality control [14], communications [51], molecular biology [4, 36], pattern matching [31, 12], and machine learning [54].

Group testing can be subdivided into two paradigms: *adaptive* and *non-adaptive*. In adaptive GT, tests are conducted sequentially, and previous results can influence the selection of subsequent groups, whereas, in the non-adaptive setting, the complete testing strategy is provided up-front. A second distinction is whether test results are perturbed by evaluation noise. In the noisy setting, there is a risk that testing a group with active elements would show a negative outcome and vice versa. Our method presented in Section 3 can be considered an adaptation of noisy adaptive GT [43].

[14] present a *Bayesian Sequential Experimental Design* approach for binary outcomes, which at each iteration selects groups that maximize one of two criteria: the first one is the mutual information between the elements' probability of being active, $\boldsymbol{\xi}$, in the selected group and the observation. The second is the area under the marginal encoder's curve (AUC). As the distribution over the active group $p(\boldsymbol{\xi})$ is a 2^n-dimensional vector, it quickly becomes impractical to store and update. Consequently, they propose using a sequential Monte Carlo (SMC) sampler [15], representing the posterior probabilities by a number of weighted particles.

3 Group testing for Bayesian optimization

Our proposed method, `GTBO`, leverages the assumption of axis-aligned active subspaces by explicitly identifying the active dimensions. This gives the user additional insight into the problem and improves sample efficiency by focusing the optimization on the active dimensions. This section describes how we adapt the GT methodology to find active dimensions in as few evaluations as possible. Subsequently, we use the information to set strong priors for the GP length scales, providing the surrogate model with the knowledge about which features are active.

Noisy adaptive group testing. The underlying assumption is that a population of n elements exists, each of which either possesses or lacks a specific characteristic. We refer to the subset of elements with this characteristic as the active group, considering the elements belonging to this group as active. We let the random variable (RV) ξ_i denote whether the element i is active ($\xi_i = 1$), or inactive ($\xi_i = 0$), similar to [14] who studied binary outcomes. The state of the whole population can be written as the random vector $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_n\} \in \{0, 1\}^n$.

We aim to uncover each element's activeness by performing repeated group tests. We write $\boldsymbol{g}$ as a binary vector $\boldsymbol{g} = \{g_1, \ldots, g_n\} \in \{0, 1\}^n$, such that $g_i = 1$ signifies that element i belongs to the group. In noisy GT, the outcome of testing a group is a random event described by the RV $A(\boldsymbol{g}, \boldsymbol{\xi}) \in \{0, 1\}$. A common assumption is that the probability distribution of $A(\boldsymbol{g}, \boldsymbol{\xi})$ only depends on whether group $\boldsymbol{g}$ contains any active elements, i.e., $\boldsymbol{g}^\intercal \boldsymbol{\xi} \geqslant 1$. In this case, one can define the sensitivity $p(A(\boldsymbol{g}, \boldsymbol{\xi}) = 1 \mid \boldsymbol{g}^\intercal \boldsymbol{\xi} \geqslant 1)$ and specificity $p(A(\boldsymbol{g}, \boldsymbol{\xi}) = 0 \mid \boldsymbol{g}^\intercal \boldsymbol{\xi} = 0)$ of the test setup.

As we assume the black-box function f to be expensive to evaluate, we select groups $\boldsymbol{g}_t$ to learn as much as possible about the distribution $\boldsymbol{\xi}$ while limiting the number of iterations to $t = 1 \ldots T$, which subsequently limits the number of function evaluations. We note that the group $\boldsymbol{g}_i$ is not a RV, but is selected as part of GT iteration i.

We can identify the active variables by modifying only a few variables in the search space and observing how the objective changes. Intuitively, if the function value remains approximately constant after perturbing a subset of variables from the default configuration, it suggests that these variables are inactive. On the contrary, if a specific dimension i is included in multiple subsets and the output changes significantly upon perturbation of those subsets, this suggests that dimension i is highly likely to be active.

Unlike in the traditional GT problem, where outcomes are binary, we work with continuous, real-valued function observations. To evaluate how a group of variables affects the objective function, we first evaluate a *default* configuration in the center of the search space, $\boldsymbol{x}_{\text{def}}$, and then vary the variables in the group and study the difference. We use the group notation $\boldsymbol{g}_t \in \{0,1\}^D$ as a binary indicator denoting which variables we change in iteration t. Similarly, we reuse the notation that the RV $\boldsymbol{\xi}$ denotes the active dimensions, and the true state is denoted by $\boldsymbol{\xi}^*$.

The new configuration to evaluate is selected as

$$\boldsymbol{x}_t = \boldsymbol{x}_{\text{def}} \oplus (\boldsymbol{g}_t \otimes \boldsymbol{u}_t), \tag{5.1}$$

where $\boldsymbol{u}_t \in [0,1]^D$ is drawn from $\mathcal{U}(\mathbf{0},\mathbf{1})$ until each active dimension has a distance of at least 0.4 to $\boldsymbol{x}_{\text{def}}$, $\oplus$ is element-wise addition, and $\otimes$ is element-wise multiplication. Note that a point $\boldsymbol{x}_t$ is always associated with a group $\boldsymbol{g}_t$ that determines along which dimensions $\boldsymbol{x}_t$ differs from the default configuration. For the newly obtained configuration $\boldsymbol{x}_t$, we must assess whether $|f(\boldsymbol{x_t}) - f(\boldsymbol{x}_{\text{def}})| \gg 0$, which would indicate that the group $\boldsymbol{g}_t$ contains active dimensions, i.e., $\boldsymbol{g}_t^\intercal \boldsymbol{\xi}^* \geqslant 1$. However, as we generally do not have access to the true values $f(\boldsymbol{x}_{\text{def}})$ or $f(\boldsymbol{x}_t)$ due to observation noise, we use an estimate $\hat{f}(\boldsymbol{x})$.

Since f can only be observed with Gaussian noise of unknown variance σ_n^2, there is always a non-zero probability that a high difference in function value occurs between $\boldsymbol{x}$ and $\boldsymbol{x}_{\text{def}}$ even if group $\boldsymbol{g}$ contains no active dimensions. Therefore, we take a probabilistic approach, which relies on two key assumptions:

1. $Z_t \coloneqq \hat{f}(\boldsymbol{x}_t) - \hat{f}(\boldsymbol{x}_{\text{def}}) \sim \mathcal{N}(0, \sigma_n^2)$ if $\boldsymbol{g}_t^\intercal \boldsymbol{\xi} = 0$, i.e., function values follow the noise distribution if the group $\boldsymbol{g}_t$ contains no active dimensions.
2. $Z_t \coloneqq \hat{f}(\boldsymbol{x}_t) - \hat{f}(\boldsymbol{x}_{\text{def}}) \sim \mathcal{N}(0, \sigma^2)$ if $\boldsymbol{g}_t^\intercal \boldsymbol{\xi} \geqslant 1$, i.e., function values are drawn from a zero-mean Gaussian distribution with the function-value variance if the group $\boldsymbol{g}_t$ contains active dimensions.

The first assumption follows from the assumption of Gaussian observation noise and an axis-aligned active subspace. The second assumption follows from a GP prior assumption on f, under which $\hat{f}(x_t)$ is normally distributed. As we are only interested in the change from $f(\boldsymbol{x}_{\text{def}})$, we assume this distribution to have mean zero.

We estimate the noise variance, σ_n^2, and function-value variance, σ^2, based on an assumption on the maximum number of active variables. First we evaluate f at the default configuration $\boldsymbol{x}_{\text{def}}$. We then split the dimensions into several roughly

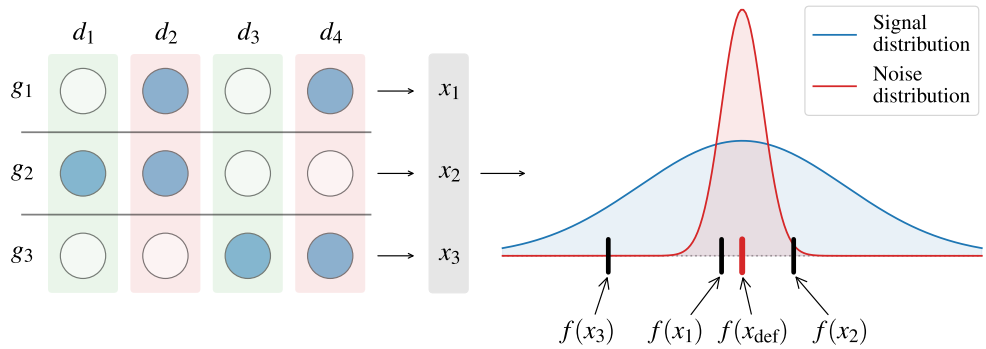

Fig. 5.1: `GTBO` assumes an axis-aligned subspace. A point x_1 that only varies along inactive dimensions (d_2 and d_4) obtains a similar function value as the default configuration (x_{def}). Points x_2 and x_3 that vary along active dimensions (d_1 and d_3) have a higher likelihood under the signal distribution than under the noise distribution.

equally sized bins. For each bin, we evaluate f on the default configuration perturbed along the direction of all variables in that bin and compare the result with the default value. We then estimate the function variance as the empirical variance among the `max_act` largest such differences and the noise variance as the empirical variance among the rest. Here, `max_act` represents the assumed maximum number of active dimensions. If the assumption holds, there can be no active dimensions in the noise estimate, which is more sensitive to outliers. It must be an upper bound, as the method is more sensitive to estimating the noise from active dimensions than vice versa.

Under Assumptions 1 and 2, the distribution of Z_t depends only on whether $\boldsymbol{g}_t$ contains active variables. Given the probability distribution over population states $p(\boldsymbol{\xi})$, the probability that $\boldsymbol{g}_t$ contains any active elements is

$$p(\boldsymbol{g}_t^\intercal \boldsymbol{\xi} \geqslant 1) = \sum_{\boldsymbol{\xi} \in \{0,1\}^D} \delta_{\boldsymbol{g}_t^\intercal \boldsymbol{\xi} \geqslant 1} p(\boldsymbol{\xi}). \tag{5.2}$$

We exemplify this in Fig. 5.1. Here, three groups are tested sequentially, out of which the second and third contain active variables. The three corresponding configurations, x_1, x_2, and x_3, give three function values shown on the right-hand side. As observing $f(x_1)$ is more likely under the noise distribution, g_1 has a higher probability of being inactive. Similarly, as $f(x_2)$ and $f(x_3)$ are more likely to be observed under the signal distribution, g_2 and g_3 are more likely to be active.

Estimating the group activeness probability. Equation (5.2) requires summing over 2^D possible activity states, which, for higher-dimensional functions, becomes prohibitively expensive. Instead, we use an SMC sampler with M particles $\{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M\}$ and particle weights $\{\omega_1, \ldots, \omega_M\}$. Each particle $\boldsymbol{\xi}_k \in \{0,1\}^D$ represents a possible ground truth. We follow the approach presented in [14] and use a modified Gibbs kernel for discrete spaces [29]. We then estimate the probability $p(\boldsymbol{g}_t^\intercal \boldsymbol{\xi} \geqslant 1)$ of a group $\boldsymbol{g}_t$ to be active by

$$\hat{p}(\boldsymbol{g}_t^\intercal \boldsymbol{\xi} \geqslant 1) = \sum_{k=1}^{M} \omega_k \delta_{\boldsymbol{g}_t^\intercal \boldsymbol{\xi}_k \geqslant 1}. \tag{5.3}$$

Choice of new groups. We choose new groups to maximize the information obtained about $\boldsymbol{\xi}$ when observing Z_t. This can be achieved by maximizing their mutual information (MI). Under Assumptions 1 and 2, we can write the MI as

$$\begin{aligned}
I(\boldsymbol{\xi}, Z_t) &= H(Z_t) - H(Z_t|\boldsymbol{\xi}) && (5.4)\\
&= H(Z_t) - \sum_{\bar{\boldsymbol{\xi}} \in \{0,1\}^D} p(\bar{\boldsymbol{\xi}}) H(Z_t|\boldsymbol{\xi} = \bar{\boldsymbol{\xi}}) && (5.5)\\
&= H(Z_t) - [p(\boldsymbol{g}_t^\intercal \boldsymbol{\xi} \geqslant 1) H(Z_t|\boldsymbol{g}_t^\intercal \boldsymbol{\xi} \geqslant 1)\\
&\quad + p(\boldsymbol{g}_t^\intercal \boldsymbol{\xi} = 0) H(Z_t|\boldsymbol{g}_t^\intercal \boldsymbol{\xi} = 0)] && (5.6)\\
&= H(Z_t) - \frac{1}{2}[p(\boldsymbol{g}_t^\intercal \boldsymbol{\xi} = 0) \log(2\sigma_n^2 \pi e)\\
&\quad + p(\boldsymbol{g}_t^\intercal \boldsymbol{\xi} \geqslant 1) \log(2\sigma^2 \pi e)]. && (5.7)
\end{aligned}$$

Since Z_t is modeled as a Gaussian mixture model (GMM), its entropy $H(Z_t)$ has no known closed-form expression [24], but can be approximated using Monte Carlo:

$$H(Z_t) = \mathbb{E}[-\log p(Z_t)] \approx -\frac{1}{N} \sum_{i=1}^{N} \log p(z_t^i), \tag{5.8}$$

and $z_t^i \sim \mathcal{N}(0, \sigma^2)$ with probability $\hat{p}(\boldsymbol{g}_t^\intercal \boldsymbol{\xi} \geqslant 1)$ and $z_t^i \sim \mathcal{N}(0, \sigma_n^2)$ with probability $\hat{p}(\boldsymbol{g}_t^\intercal \boldsymbol{\xi} = 0)$.

Maximizing the mutual information. `GTBO` optimizes the MI using a multi-start forward-backward algorithm [42]. First, several initial groups are generated by sampling from the prior and the posterior over $\boldsymbol{\xi}$. Then, elements are greedily added for each group in a *forward phase* and removed in a subsequent *backward*

phase. In the forward phase, we incrementally include the element that results in the greatest MI increase. Conversely, in the backward phase, we eliminate the element that contributes the most to MI increase. Each phase is continued until no further elements are added or removed from the group. Finally, the group with the largest MI is returned.

Batch evaluations. If the black-box function can be run in parallel, we greedily select additional groups by running the forward-backward algorithm again, excluding already selected groups. For high-dimensional problems there are frequently several distinct groups which each yields close to optimal MI. We continue adding groups to evaluate until we have reached a user-specified upper limit or until the MI of new groups drops below a threshold.

Updating the activeness probability. Once we have selected a new group $\boldsymbol{g}_t$ and observed the corresponding function value z_t, we update our estimate of $\hat{p}(\boldsymbol{\xi}_k)$ for each particle k:

$$\hat{p}^t(\boldsymbol{\xi}_k) \propto \hat{p}^{t-1}(\boldsymbol{\xi}_k) p(z_t|\boldsymbol{\xi}_k) \tag{5.9}$$

$$\propto \hat{p}^{t-1}(\boldsymbol{\xi}_k) \begin{cases} p(z_t|\boldsymbol{g}_t^\intercal \boldsymbol{\xi}_k \geqslant 1) & \text{if } \boldsymbol{g}_t^\intercal \boldsymbol{\xi}_k \geqslant 1 \\ p(z_t|\boldsymbol{g}_t^\intercal \boldsymbol{\xi}_k = 0) & \text{if } \boldsymbol{g}_t^\intercal \boldsymbol{\xi}_k = 0, \end{cases} \tag{5.10}$$

where $p(z_t|\boldsymbol{g}_t^\intercal \boldsymbol{\xi}_k = 0)$ and $p(z_t|\boldsymbol{g}_t^\intercal \boldsymbol{\xi}_k \geqslant 1)$ are Gaussian likelihoods. Assuming that the probabilities of dimensions to be active are independent, the prior probability is given by $\hat{p}^0(\boldsymbol{\xi}_k) = \prod_{i=1}^D q_i^{\boldsymbol{\xi}_{k,i}} (1-q_i)^{1-\boldsymbol{\xi}_{k,i}}$ where q_i is the prior probability for the i-th dimension to be active. As we represent the probability distribution $\hat{p}^0(\boldsymbol{\xi})$ by a point cloud, any prior distribution can be used to insert prior knowledge. We use the same SMC sampler as [14].

The GTBO algorithm. With the individual parts defined, we present the complete procedure for `GTBO`. `GTBO` iteratively selects and evaluates groups for T iterations or until convergence. We consider it to have converged when the posterior marginal probability for each variable $\hat{p}^t(\xi_i)$ lies in $[0, C_{\text{lower}}] \cup [C_{\text{upper}}, 1]$, for some convergence thresholds C_{lower} and C_{upper}.

Subsequently, their marginal posterior distribution decides which variables are selected to be active. A variable i is considered active if its marginal is larger than some threshold, $\hat{p}_i^t(\boldsymbol{\xi}) \geqslant \eta$. Once we have deduced which variables are active, we perform BO using the remaining sample budget. To strongly focus on the active subspace, we use short lengthscale priors for the active variables and

long lengthscale priors for the inactive variables. We use a GP with a Matérn$-5/2$ kernel as the surrogate model and `qLogNoisyExpectedImprovement` [3] as the acquisition function. The BO phase is initialized with data sampled during the feature selection phase. Several points are sampled throughout the GT phase that only differ marginally in the active subspace. Such duplicates are removed to facilitate the fitting of the GP.

4 Computational experiments

In this section, we showcase the performance of the proposed methodology, both for finding the relevant dimensions and for the subsequent optimization. We compare state-of-the-art frameworks for high-dimensional BO on several synthetic and real-life benchmarks. `GTBO` outperforms previous approaches on the tested real-world and synthetic benchmarks. In Section 4.2, we study the sensitivity of `GTBO` to external traits of the optimization problem, such as noise-to-signal ratio and the number of active dimensions. The code for `GTBO` is available at `https://github.com/gtboauthors/gtbo`.

4.1 Experimental setup

We test `GTBO` on four synthetic benchmark functions, `Branin2`, `Levy` in 4 dimensions, `Hartmann6`, and `Griewank` in 8 dimensions, which we extend with inactive "dummy" dimensions [50, 19, 37] as well as two real-world benchmarks: the 124D soft-constraint version of the `Mopta08` benchmark [19], and the 180D `LassoDNA` benchmark [44]. We add significant observation noise for the synthetic benchmarks, but the inactive dimensions are truly inactive. In contrast, the real-world benchmarks do not exhibit observation noise, but all dimensions have at least a marginal impact on the objective function. Note that the noisy synthetic benchmarks are considerably more challenging for `GTBO` than their noiseless counterparts.

Since the search space center is a decent solution for `LassoDNA`, `GTBO` chooses a default configuration for each GT repetition uniformly at random. To not give `BAxUS` a similar advantage, we subtract a random offset from the search space bounds, which we add again before evaluating the function. This ensures that `BAxUS` cannot always represent the near-optimal origin.

To evaluate the BO performance, we benchmark against `TuRBO` [20] with one and five trust regions, `SAASBO` [19], `CMA-ES` [22], `HeSBO` [34], and `BAxUS` [37]

using the implementations and settings provided by the authors, unless stated otherwise. We compare against random search, i.e., choose points in the search space uniformly at random.

We use the `pycma` implementation for `CMA-ES` [23] and the `Ax` implementation for `Alebo` [2]. To show the effect of different choices of the target dimensionality d, we run `Alebo` with $d = 10$ and $d = 20$. We observed that `Alebo` and `SAASBO` are constrained by their high runtime and memory consumption. The available hardware allowed up to 100 evaluations for `SAASBO` and 300 evaluations for `Alebo` for each run. Larger sampling budgets or higher target dimensions for `Alebo` resulted in out-of-memory errors. We note that limited scalability was expected for these two methods, whereas the other methods scaled to considerably larger budgets, as required for scalable BO. We initialize each optimizer with ten initial samples and `BAxUS` with $b = 3$ and $m_D = 1000$ and run ten repeated trials. Plots show the mean logarithmic regret for synthetic benchmarks and the mean function value for real-world benchmarks. The shaded regions indicate one standard error.

Unless stated otherwise, we run `GTBO` with 10 000 particles for the SMC sampler, the prior probability of being active, $q = 0.05$, and 3 initial groups for the forward-backward algorithm. When estimating the function signal and noise variance, we set the assumed maximum number of active dimensions, `max_act`, to $\sqrt{D}$. The threshold to be considered active after the GT phase, η, is set to 0.5, and the lower and upper convergence thresholds, C_{lower} and C_{upper}, are $5 \cdot 10^{-3}$ and 0.9. We run all experiments with a log-normal $\mathcal{LN}(7, 1)$ length scale prior to the inactive dimensions. If a benchmark is known to have strictly active and inactive parameters, this prior can be chosen more aggressively to "switch off" the inactive dimensions. We use a $\mathcal{LN}(0, 1)$ prior for the active variables, resulting in significantly shorter length scales. In the GT phase, we use batch evaluation with a maximum of 5 groups in each batch and a maximum MI drop of 1%. Note that we still count the number of evaluations, not the number of batches, towards the budget. The experiments are run on Intel Xeon Gold 6130 machines using two cores.

4.2 Performance of the group testing

Before studying `GTBO`'s overall optimization performance in high-dimensional settings, we analyze the performance of the GT procedure. In Fig. 5.2, we show the evolution of the average marginal probability of being active over the iterations for the different dimensions. The truly active dimensions are plotted in green, and the inactive ones are in blue squares. For all the problems, `GTBO`

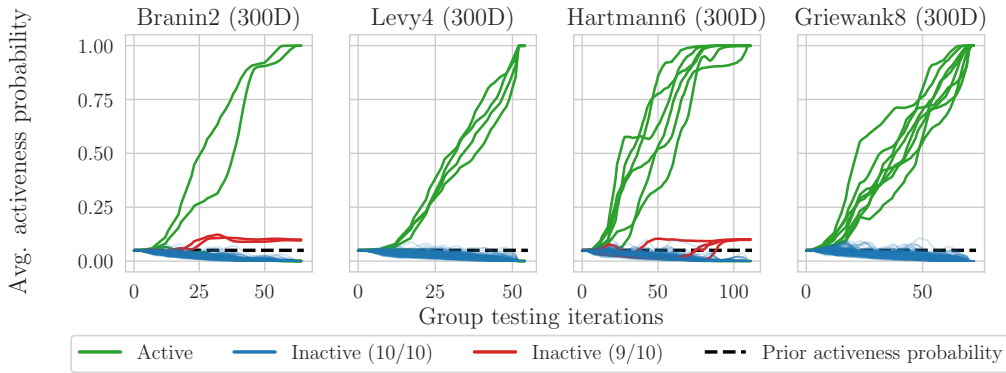

Fig. 5.2: Evolution of the average marginal probability of being active across ten repetitions. Each line represents one dimension; active dimensions are colored green, and inactive dimensions are blue. In the few cases where GTBO finds inactive variables to be active, the lines are emphasized in red. The last iteration marks the end of the *longest* GT phase across all runs. All active dimensions are identified in all runs. 6 out of 1180 inactive dimensions are incorrectly classified as active *once* in ten runs across the benchmarks, implying a false positive rate of slightly above 0.05%.

correctly classifies all active dimensions during all runs within 39–112 iterations. Across ten runs, GTBO misclassifies 6 out of 1180 inactive variables to be active once each, for a false positive rate of 0.05%.

Sensitivity analysis. We explore the sensitivity of GTBO to the output noise and problem size by evaluating it on the Levy4 synthetic benchmark extended to 100 dimensions, with a noise standard deviation of 0.1, and varying the properties of interest. In Fig. 5.3, we show how the percentage of correctly predicted variables evolves with the number of tests t for different functional properties. Correctly classified is defined here as having a probability of less than 1% if inactive or above 90% if active. GTBO shows to be robust to lower noise but suffers from very high noise levels. As expected, higher function dimensionality and number of active dimensions increases the time until convergence. Note that the signal and noise variance estimates build on the assumption that there are a maximum of $\sqrt{D}$ active dimensions, which does not hold with 32 active dimensions.

4.3 Optimization of real-world and synthetic benchmarks

We show that identifying the relevant variables can drastically improve optimization performance. Fig. 5.5 shows the performance of GTBO and competitors on the real-world benchmarks, Fig. 5.4 on the synthetic benchmarks. The results show

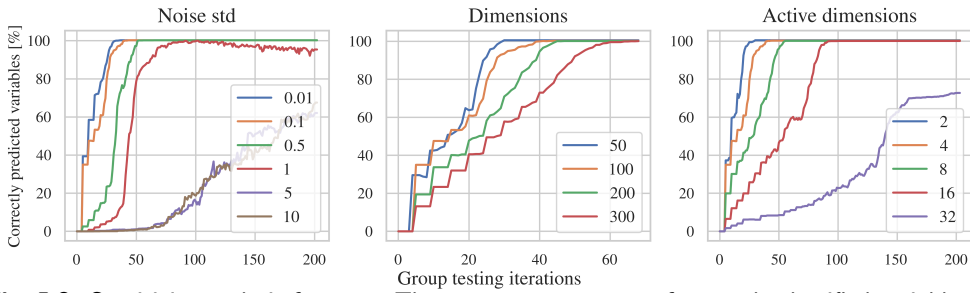

Fig. 5.3: Sensitivity analysis for `GTBO`. The average percentage of correctly classified variables is displayed for increasing GT iterations. The percentage is ablated for (left) various levels of *output noise*, (middle) number of *total dimensions*, and (right) number of *effective dimensions*. Each legend shows the configurations of the respective parameter.

the incumbent function value for each method, averaged over ten repeated trials. We plot the true average incumbent function values on the noisy benchmarks without observation noise.

Note that `Griewank` has its optimum in the center of the search space. To not gain an unfair advantage, we run `GTBO` with a non-standard default away from the optimum. However, the optimum being in the center means that all linear projections will contain the optimum, which boosts the projection-based methods `Alebo` and `BAxUS` as it allows them to represent the optimum regardless of the embedding choice.

Figure 5.5 shows `GTBO`'s performance on the 124D `Mopta08` and 180D `LassoDNA` benchmarks. On `Mopta08`, `GTBO` performs like a random search during the GT phase but quickly outperforms the other methods once the BO phase begins. This behavior suggests that several dimensions of the `Mopta08` benchmark have negligible impact on the optimization objective and highlights `GTBO`'s ability to leverage the activeness information for benchmarks with inactive dimensions efficiently.

On `LassoDNA`, `GTBO` again reaches similar levels of performance as the random search and improves significantly when the BO phase starts, outperforming `CMA-ES` after 1 000 function evaluations. However, the overall performance is not on par with other BO methods. Perhaps the fact that both variable selection methods, `MCTS-VS` and `GTBO`, fail to reach the same levels of performance as `TuRBO` and `BAxUS` suggests that most dimensions in `LassoDNA` are active. While this benchmark is suspected to violate our axis-aligned assumptions, `GTBO` still shows reasonable performance and outperforms `MCTS-VS` by a large margin.

Overall, `GTBO` first identifies relevant variables, followed by a sharp drop when

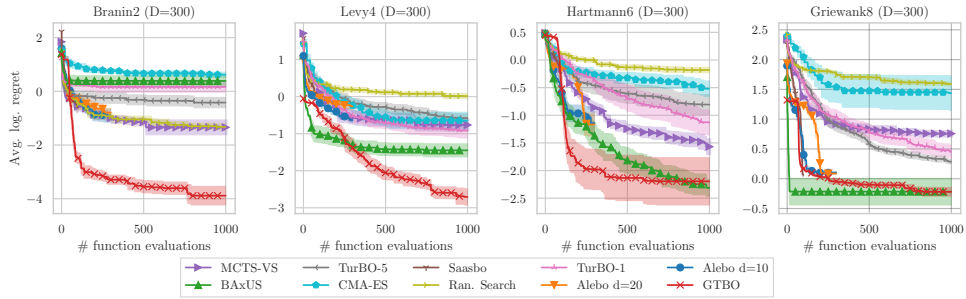

Fig. 5.4: `GTBO` finds active dimensions and optimizes efficiently on synthetic noisy benchmarks (`Branin2`, `Levy4`, `Hartmann6`, and `Griewank8`).

the optimization phase starts, indicating that knowing the active dimensions drastically speeds up optimization. In the real-world `Mopta08` application where the dimensions detected as inactive have negligible impact on the objective, `GTBO` outperforms state-of-the-art methods despite the delayed onset of the BO phase. However, `GTBO` can suffer in cases where the axis-alignment assumption does not hold, as shown by the `LassoDNA` results.

5 Discussion

Optimizing expensive-to-evaluate high-dimensional black-box functions is a challenge for applications in industry and academia. We propose `GTBO`, a novel algorithm that focuses the Bayesian optimization on variables found relevant in a preceding group testing phase. `GTBO` explicitly exploits the structure of a sparse axis-aligned subspace to reduce the complexity of an application in high dimensions and is the first method to adapt group testing, in which it aims to find infected individuals by conducting pooled tests, to Bayesian optimization. It differs from `SAASBO` [19] in that it yields a clear-cut decision on which variables are active or inactive and from `Alebo` [28] or `BAxUS` [37] in that it does not rely on random projections to identify relevant variables. Similarly to `MCTS-VS` [47], our method works by explicitly identifying the set of relevant variables; however, `GTBO` is the first method to use a more principled approach to learn them by employing Group Testing principles and theory.

`GTBO` quickly detects active and inactive variables and shows robust optimization performance in synthetic and real-world settings. Furthermore, the GT phase yields a set of relevant dimensions, which allows users to learn something fundamental about their application. For example, on the `Mopta08` benchmark, the user learns which shape parameters minimize vehicle mass under some con-

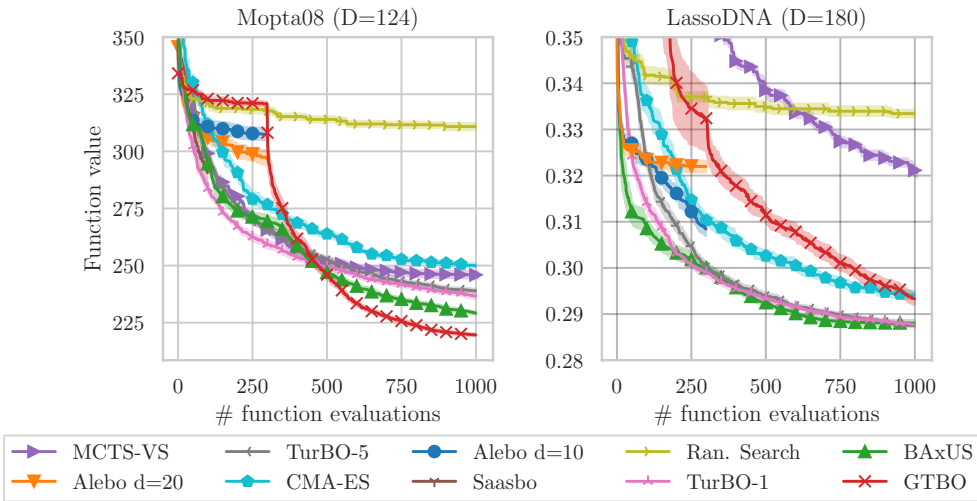

Fig. 5.5: `GTBO` outperforms competitors in real-world experiments. Notably, the performance on `Mopta08` increases significantly after the GT phase at iteration 300, suggesting that the dimensions found during the GT phase are highly relevant. For `LassoDNA`, the performance is worse for both `GTBO` and `MCTS-VS`, indicating that the assumption of an axis-aligned subspace is violated.

straints [25] `GTBO` robustly uses the activeness information so that it can still optimize efficiently even if the inactive dimensions have a marginal impact on the objective function.

In future work, we will fuse the GT and BO phases so that the non-default values in a group test guide the optimization. This will further increase the sample efficiency of `GTBO`. For problems where inactive dimensions do not have any effect on the objective function, `GTBO` can be used to identify the active dimensions and then only optimize on those. This further improves sample efficiency and is advantageous for applications where optimizing over a larger set of dimensions incurs additional costs [30].

Limitations. `GTBO` relies on the assumption that an application has several irrelevant parameters. If this assumption is unmet, the method might underperform or waste a fraction of the evaluation budget to identify all variables as relevant. Furthermore, `GTBO` cannot exploit problems with a low-dimensional subspace that is not axis-aligned, and it is unclear how many tests are required for `GTBO`'s GT phase to converge as it is challenging to prove bounds in the non-asymptotic regime [14].

Broader impact

This paper presents basic research with the goal of advancing the fields of Machine Learning and optimization. There are many potential societal impacts of our work, none of which should be specifically highlighted here.

References

[1] Matthew Aldridge, Oliver Johnson, Jonathan Scarlett, et al. Group testing: an information theory perspective. *Foundations and Trends® in Communications and Information Theory*, 15(3-4):196–392, 2019.

[2] Eytan Bakshy, Lili Dworkin, Brian Karrer, Konstantin Kashin, Benjamin Letham, Ashwin Murthy, and Shaun Singh. AE: A domain-agnostic platform for adaptive experimentation. In *Conference on Neural Information Processing Systems*, pages 1–8, 2018.

[3] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. In *Advances in Neural Information Processing Systems*, 2020. URL `http://arxiv.org/abs/1910.06403`.

[4] DJ Balding, WJ Bruno, DC Torney, and E Knill. A comparative survey of non-adaptive pooling designs. In *Genetic mapping and DNA sequencing*, pages 133–154. Springer, 1996.

[5] Archit Bansal, Danny Stoll, Maciej Janowski, Arber Zela, and Frank Hutter. JAHS-Bench-201: A Foundation For Research On Joint Architecture And Hyperparameter Search. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[6] Felix Berkenkamp, Andreas Krause, and Angela Schoellig. Bayesian Optimization with Safety Constraints: Safe and Automatic Parameter Tuning in Robotics. *Machine Learning*, 06 2021. doi: 10.1007/s10994-021-06019-1.

[7] Mickael Binois and Nathan Wycoff. A survey on high-dimensional gaussian process modeling with application to bayesian optimization. *ACM Transactions on Evolutionary Learning and Optimization*, 2(2):1–26, 2022.

[8] Mohamed Amine Bouhlel, Nathalie Bartoli, Abdelkader Otsmane, and Joseph Morlier. Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction. *Structural and Multidisciplinary Optimization*, 53:935–952, 2016.

[9] Benjamin Burger, Phillip M Maffettone, Vladimir V Gusev, Catherine M Aitchison, Yang Bai, Xiaoyan Wang, Xiaobo Li, Ben M Alston, Buyi Li, Rob Clowes, et al. A mobile robotic chemist. *Nature*, 583(7815):237–241, 2020.

[10] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. Deisenroth. Bayesian gait optimization for bipedal locomotion. In P. Pardalos and M. Resende, editors, *Proceedings of the Eighth International Conference on Learning and Intelligent Optimization (LION'14)*, Lecture Notes in Computer Science. Springer, 2014.

[11] Yutian Chen, Aja Huang, Ziyu Wang, Ioannis Antonoglou, Julian Schrittwieser, David Silver, and Nando de Freitas. Bayesian optimization in alphago. *CoRR*, abs/1812.06855, 2018. URL `http://arxiv.org/abs/1812.06855`.

[12] Raphaël Clifford, Klim Efremenko, Ely Porat, and Amir Rothschild. Pattern matching with don't cares and few errors. *Journal of Computer and System Sciences*, 76(2):115–124, 2010.

[13] Paul G Constantine, Armin Eftekhari, and Michael B Wakin. Computing active subspaces efficiently with gradient sketching. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 353–356. IEEE, 2015.

[14] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Noisy Adaptive Group Testing using Bayesian Sequential Experimental Design. *CoRR*, abs/2004.12508, 2020.

[15] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[16] Josip Djolonga, Andreas Krause, and Volkan Cevher. High-dimensional gaussian process bandits. *Advances in neural information processing systems*, 26, 2013.

[17] Robert Dorfman. The detection of defective members of large populations. *The Annals of mathematical statistics*, 14(4):436–440, 1943.

[18] Adel Ejjeh, Leon Medvinsky, Aaron Councilman, Hemang Nehra, Suraj Sharma, Vikram Adve, Luigi Nardi, Eriko Nurvitadhi, and Rob A Rutenbar. HPVM2FPGA: Enabling True Hardware-Agnostic FPGA Programming. In *Proceedings of the 33rd IEEE International Conference on Application-specific Systems, Architectures, and Processors*, 2022.

[19] David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pages 493–503. PMLR, 2021.

[20] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

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

[22] Nikolaus Hansen and Andreas Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE international conference on evolutionary computation*, pages 312–317. IEEE, 1996.

[23] Nikolaus Hansen, yoshihikoueno, ARF1, Kento Nozawa, Luca Rolshoven, Matthew Chan, Youhei Akimoto, brieglhostis, and Dimo Brockhoff. CMA-ES/pycma: r3.2.2. March 2022.

[24] Marco F Huber, Tim Bailey, Hugh Durrant-Whyte, and Uwe D Hanebeck. On entropy approximation for Gaussian mixture random vectors. In *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 181–188. IEEE, 2008.

[25] Donald R Jones. Large-scale multi-disciplinary mass optimization in the auto industry. In *MOPTA 2008 Conference (20 August 2008)*, 2008.

[26] Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Neural architecture search with bayesian optimisation and optimal transport. *Advances in neural information processing systems*, 31, 2018.

[27] B. Letham, K. Brian, G. Ottoni, and E. Bakshy. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 2018.

[28] Ben Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional Bayesian optimization. *Advances in neural information processing systems*, 33:1546–1558, 2020.

[29] Jun S Liu. Peskun's theorem and a modified discrete-state Gibbs sampler. *Biometrika*, 83(3), 1996.

[30] Sulin Liu, Qing Feng, David Eriksson, Benjamin Letham, and Eytan Bakshy. Sparse bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 3754–3774. PMLR, 2023.

[31] Anthony J Macula and Leonard J Popyack. A group testing method for finding patterns in data. *Discrete applied mathematics*, 144(1-2):149–157, 2004.

[32] Matthias Mayr, Carl Hvarfner, Konstantinos Chatzilygeroudis, Luigi Nardi, and Volker Krueger. Learning skill-based industrial robot tasks with user priors. *IEEE 18th International Conference on Automation Science and Engineering*, 2022. URL `https://arxiv.org/abs/2208.01605`.

[33] L. Nardi, D. Koeplinger, and K. Olukotun. Practical design space exploration. In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 347–358. IEEE, 2019.

[34] Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*, pages 4752–4761. PMLR, 2019.

[35] Diana M Negoescu, Peter I Frazier, and Warren B Powell. The knowledge-gradient algorithm for sequencing experiments in drug discovery. *INFORMS Journal on Computing*, 23(3):346–363, 2011.

[36] Hung Q Ngo and Ding-Zhu Du. A survey on combinatorial group testing algorithms with applications to DNA library screening. *Discrete mathematical problems with medical applications*, 55:171–182, 2000.

[37] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Increasing the scope as you learn: Adaptive bayesian optimization in nested subspaces. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=e4Wf6112DI`.

[38] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Bounce: a Reliable Bayesian Optimization Algorithm for Combinatorial and Mixed Spaces. In *Advances in Neural Information Processing Systems*, 2023.

[39] Giulia Pedrielli and Szu Hui Ng. G-STAR: A new kriging-based trust region method for global optimization. In *2016 Winter Simulation Conference (WSC)*, pages 803–814. IEEE, 2016.

[40] Rommel G Regis. Trust regions in Kriging-based optimization with expected improvement. *Engineering optimization*, 48(6):1037–1059, 2016.

[41] Binxin Ru, Xingchen Wan, Xiaowen Dong, and Michael Osborne. Interpretable neural architecture search via bayesian optimisation with weisfeiler-lehman kernels. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=j9Rv7qdXjd`.

[42] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[43] Jonathan Scarlett. Noisy adaptive group testing: Bounds and algorithms. *IEEE Transactions on Information Theory*, 65(6):3646–3661, 2018.

[44] Kenan Šehić, Alexandre Gramfort, Joseph Salmon, and Luigi Nardi. LassoBench: A High-Dimensional Hyperparameter Optimization Benchmark Suite for Lasso. *arXiv preprint arXiv:2111.02790*, 2021.

[45] B. Shahriari, A. Bouchard-Cote, and N. de Freitas. Unbounded Bayesian optimization via regularization. In A. Gretton and C. Robert, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 51, pages 1168–1176. Proceedings of Machine Learning Research, 2016.

[46] Yihang Shen and Carl Kingsford. Computationally Efficient High-Dimensional Bayesian Optimization via Variable Selection. In *AutoML Conference 2023*, 2023.

[47] Lei Song, Ke Xue, Xiaobin Huang, and Chao Qian. Monte carlo tree search based variable selection for high dimensional bayesian optimization. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=SUzPos_pUC`.

[48] Doniyor Ulmasov, Caroline Baroukh, Benoit Chachuat, Marc Peter Deisenroth, and Ruth Misener. Bayesian optimization with dimension scheduling: Application to biological systems. In *Computer Aided Chemical Engineering*, volume 38, pages 1051–1056. Elsevier, 2016.

[49] Xingchen Wan, Vu Nguyen, Huong Ha, Binxin Ru, Cong Lu, and Michael A Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. *International Conference on Machine Learning (ICML) 38*, 2021.

[50] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

[51] Jack Wolf. Born again group testing: Multiaccess communications. *IEEE Transactions on Information Theory*, 31(2):185–191, 1985.

[52] Nathan Wycoff, Mickael Binois, and Stefan M Wild. Sequential learning of active subspaces. *Journal of Computational and Graphical Statistics*, 30(4): 1224–1237, 2021.

[53] Yichi Zhang, Daniel W Apley, and Wei Chen. Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Scientific reports*, 10(1):1–13, 2020.

[54] Yingbo Zhou, Utkarsh Porwal, Ce Zhang, Hung Q Ngo, XuanLong Nguyen, Christopher Ré, and Venu Govindaraju. Parallel feature selection inspired by group testing. *Advances in neural information processing systems*, 27, 2014.

Paper VI

Vanilla Bayesian Optimization Performs Great in High Dimensions

Carl Hvarfner
Lund University

Erik Hellsten
Lund University

Luigi Nardi
Lund University

Abstract

High-dimensional problems have long been considered the Achilles' heel of Bayesian optimization. Spurred by the curse of dimensionality, a large collection of algorithms aim to make it more performant in this setting, commonly by imposing various simplifying assumptions on the objective. In this paper, we identify the degeneracies that make vanilla Bayesian optimization poorly suited to high-dimensional tasks, and further show how existing algorithms address these degeneracies through the lens of lowering the model complexity. Moreover, we propose an enhancement to the prior assumptions that are typical to vanilla Bayesian optimization, which reduces the complexity to manageable levels without imposing structural restrictions on the objective. Our modification - a simple scaling of the Gaussian process lengthscale prior with the dimensionality - reveals that standard Bayesian optimization works drastically better than previously thought in high dimensions, clearly outperforming existing state-of-the-art algorithms on multiple commonly considered real-world high-dimensional tasks.

1 Introduction

In Bayesian optimization, *complexity* and *dimensionality* are intrinsically interlinked — the higher the problem dimensionality, the harder it is to optimize. The exuberance of space, and large distance between observations, makes the size of high-variance regions along the boundary of the search space exponentially large [35, 7]. Moreover, the growing number of parameters of the Gaussian Process (GP) surrogate in relation to the number of observations makes accurate modeling of the problem at hand exceedingly difficult. In recent years, the effort to create methods that achieve efficient Bayesian optimization (BO) in high dimensions has been substantial, making it one of the most frequently addressed challenges in the BO research community [28, 65, 39, 16, 15, 45, 46, 70].

While approaches are plentiful and diverse, they all share a common characteristic: they employ restrictions on the objective which reduces its a-priori *assumed complexity* by contracting the search space. This in turn decreases distances between data points and prospective queries, increasing their correlation, thus making GP inference more informative. Assuming a degree of complexity which enables meaningful correlation is essential to efficiently optimize problems of *any* dimensionality. Nevertheless, the high-complexity, low-correlation issue presents itself most clearly in the high-dimensional setting.

In this paper, we hypothesize that the shortcomings of vanilla BO in high dimensions are strictly a consequence of the complexity assumptions imposed on the objective. To that end, we view existing high-dimensional BO (HDBO) approaches through the lens of model complexity, which arises from their structural assumptions. Thereafter, we modify standard BO to follow a similarly complexity reduced structure, simply by appropriately scaling the lengthscale prior of the GP kernel. Consequently, we effectively circumvent the well-established Curse of Dimensionality (CoD) without introducing any of the conventional structural restrictions on the objective that are prevalent in HDBO. We demonstrate that standard BO works drastically better than previously thought for high-dimensional tasks, outclassing existing high-dimensional BO algorithms on a wide range of real-world problems. Further, we aim to shed light on the inner workings of the BO machinery and why minimal changes in assumptions yield a dramatic increase in performance. The result is a performant *vanilla* BO algorithm for dimensionalities well into the thousands.

Formally, we make the following contributions:

1. We demonstrate the crucial difference between dimensionality and com-

plexity in BO, highlighting the failure modes related to high *assumed complexity* and relate existing HDBO classes to a reduction in complexity.

2. We prove that when the model is uninformed, `EI` will *not* exihibit exploratory behavior along the boundary, contrasting claims of [60, 43].

3. We propose a plug-and-play enhancement to the vanilla BO algorithm that reduces the assumed complexity to enable high-dimensional optimization, and extensively validate it across a wide spectrum of dimensionalities. Results show that vanilla BO works significantly better for high-dimensional problems than previously imagined, substantially outperforming state-of-the-art HDBO methods on a wide range of real-world tasks.

2 Background

In this section, we review the background related to Gaussian processes and Bayesian optimization. We outline the maximal information gain (MIG) as a measure of problem complexity, and the model-level choices that impact the a-priori assumed problem complexity, to subsequently explore pitfalls of vanilla BO for high-complexity tasks in Sec 4.

2.1 Gaussian Processes

The Gaussian process (GP) has become the model class of choice in most BO applications. The GP provides a distribution over functions $\hat{f} \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$ fully defined by the mean function $m(\cdot)$ and the covariance function $k(\cdot,\cdot)$. Under this distribution, the value of the function $\hat{f}(\boldsymbol{x})$, at a given location $\boldsymbol{x}$, is normally distributed with a closed-form solution for the mean $\mu(\boldsymbol{x})$ and variance $\sigma^2(\boldsymbol{x})$. We model a constant mean, so that the dynamics are fully determined by the covariance function $k(\cdot,\cdot)$.

To account for differences in variable importance, each dimension is individually scaled using lengthscale hyperparameters ℓ_i. This is commonly referred to as Automatic Relevance Determination (ARD) [67]. For D-dimensional inputs $\boldsymbol{x}$ and $\boldsymbol{x'}$, the distance $r(\boldsymbol{x}, \boldsymbol{x'})$ is subsequently computed as $r^2 = \sum_{i=1}^{D}(x_i - x_i')^2/\ell_i^2$. Along with the signal variance σ_f and *noise variance* σ_ε^2, $\boldsymbol{\theta} = \{\boldsymbol{\ell}, \sigma_\varepsilon^2, \sigma_f^2\}$ comprise the set of hyperparameters that are conventionally learned, with a possible addition of a learnable constant mean c [3, 54]. The likelihood surface $p(\boldsymbol{\theta}|\mathcal{D})$ for the GP hyperparameters is typically highly multi-modal [48, 69] and

desirable hyperparameters are conventionally found by MAP estimation, where a hyperprior is set on the kernel hyperparameters $\boldsymbol{\theta}$. While often overlooked, the choice of hyperprior can greatly impact the performance of a BO algorithm in practice, particularly in non-conventional problem settings [15, 4, 50, 51, 25].

2.2 Bayesian Optimization

We aim to find a maximizer $\boldsymbol{x}^* \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$ of the black-box function $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$, over the D-dimensional input space $\mathcal{X} = [0, 1]^D$. We assume that f can only be observed point-wise and that the observations are perturbed by Gaussian noise, $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.

The *acquisition function* uses the surrogate model to quantify the utility of a point in the search space. Acquisition functions employ a trade-off between exploration and exploitation, typically using a greedy heuristic to do so. Most common is the Expected Improvement (`EI`) [26, 9] and its numerically stable, easy-to-optimize adaptation `LogEI` [1]. Another acquisition function which uses similar heuristics is the Upper Confidence Bound [59, 58].

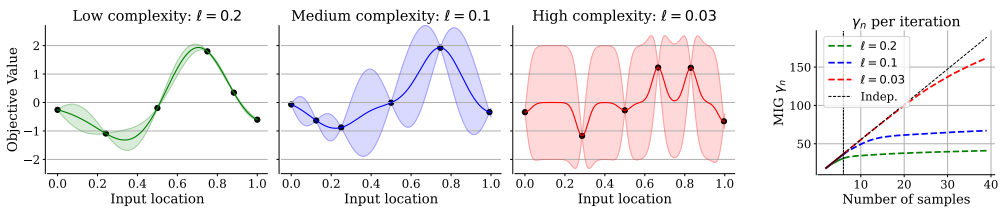

Fig. 6.1: Three models (green, blue, red) with varying lengthscales, and thus varying complexity, attempting to model the same objective, acquiring data by greedily maximizing the IG. The MIG is shown for the three models as well as an independent kernel (dashed black), where the matrix $\mathbf{K} = \boldsymbol{I}$. The MIG for the complex model closely follows the independent kernel for 20 samples, suggesting that the complex model can acquire 20 data of approximately maximal variance. The vertical line in the MIG-plot indicates the current iteration.

A Working Definition of "Vanilla" BO

In addition to the two main components — the probabilistic surrogate model and the acquisition function - BO entails multiple hidden design choices, that are paramount to its efficiency. We consider the vanilla BO algorithm to standardize the output values, and to use either a Squared Exponential (Radial Basis Function, RBF) [27] or a $\frac{5}{2}$-Matérn [54, 56] Kernel, with ARD lengthscales, an `EI`-family acquisition function and multi-start gradient-based acquisition function optimization.

While both MLE and MAP are commonly used for hyperparameter selection in practical Bayesian optimization, prevalent BO frameworks [3, 19, 24] employ MAP estimation, setting a prior on $p(\boldsymbol{\theta})$. While not included in our definition of the vanilla algorithm, the prior $p(\boldsymbol{\ell})$ commonly places high density on low values of $\boldsymbol{\ell}$. Furthermore, broad, uninformative priors are conventionally used on σ_ε^2 and σ_f^2. While fully Bayesian hyperparameter treatment [44, 54] may also be used, we do not consider it part of the vanilla algorithm.

2.3 The Maximal Information Gain

Our work centers around the *assumed* complexity of a problem, which conceptually could be seen as the size of the space of functions that have non-negligible probability under the GP prior. To quantify the assumed complexity, we use the *Maximal Information Gain* (MIG) [58] measure, which is the maximum obtainable information about the function from querying a fixed number of points. Firstly, we recall the *Information Gain* (IG) for a GP model and a set of points $\boldsymbol{X}$ is defined as

$$I(y_{\boldsymbol{X}}, f_{\boldsymbol{X}}) = \frac{1}{2} \log |\boldsymbol{I} + \sigma_\varepsilon^{-2} \mathbf{K}|, \tag{6.1}$$

where $\mathbf{K} = k(\boldsymbol{X}, \boldsymbol{X})$ is the Gram matrix for $\boldsymbol{X}$. Then, for a fixed number of points $|\boldsymbol{X}_n| = n$, the MIG is the maximizer of this measure

$$\gamma_n = \max_{\boldsymbol{X}_n \subset \mathcal{X}} I(y_{\boldsymbol{X}_n}, f_{\boldsymbol{X}_n}). \tag{6.2}$$

For fixed observation noise, the MIG is fully defined by the covariance matrix, which in turn depends on the choice of kernel, the problem dimensionality and the kernel hyperparameters. The MIG is maximal when the samples are independent, i.e., $\mathbf{K} \approx \boldsymbol{I}$. The MIG lacks a closed form solution, but is approximated to $(1 - 1/e)$-accuracy by sequentially querying the set of points with maximal posterior variance [40, 32].

Since the MIG measures the assumed complexity of f, it effectively quantifies the difficulty of optimizing a given task within a Bayesian optimization context [58, 59, 5], given that *the assumptions on k are accurate*. As long as the the MIG is nearly linear in the number of observations, there are regions in the search space that are almost independent of the collected data under the model. As such, there are still locations that we know nothing about, which makes optimization difficult. On the contrary, a small growth rate of the MIG suggests that the model would learn little by querying an additional point.

In Fig. 6.1, we provide an intuition for the MIG. We show a simpler model (left, green), as well as increasingly complex models (blue, red) for six data points. Their associated MIGs (right) are displayed for the current iteration (solid) and subsequent iterations (dashed) up until iteration 40. For the simpler model, there is little left to learn about the function, and as such, its subsequent MIG growth is small. *If* the green model is accurate, subsequent optimization is trivial due to efficient modelling that stems from large correlation in the data. On the contrary, the almost-zero correlation displayed in the red model makes its optimization vastly more difficult. This point is further emphasized by its MIG, which starts to deviate substantially from an independent kernel first after 20 data points. This suggests that *the model has capacity for 20 almost-maximal variance data acquisitions*, despite modelling only a one-dimensional objective.

3 Related Work

Multiple approaches have been proposed to tackle the limitations of BO in high dimensions. These resort to structural assumptions on the objective, which we outline by class.

Low-dimensional active subspace methods assume the existence of a lower-dimensional space which is representative of the function in the full-dimensional space. The active subspace can can be either axis-aligned [39, 15, 45, 46] or non-axis-aligned [18, 65, 30, 8, 33]. Explicit variable selection approaches [12, 34, 64, 57, 23] employ the axis-aligned assumption to identify important variables to optimize over.

Additive kernels [13, 28, 17, 66, 49, 21, 70] decompose the objective into a sum of low-dimensional component functions, where by assumption each component is only impacted by a small subset of all variables. As such, the maximal dimensionality of each component is substantially lower than the full dimensionality of f.

Local Bayesian optimization approaches [16, 38, 63, 41, 68] adaptively restrict the search space to combat the CoD, limiting the optimization to a subset of the search space. By focusing on a smaller portion of the search space, the model exhibits less variation than a global model, which simplifies

optimization. Moreover, enforcing local optimization decreases the susceptibility of the optimizer to the model.

Non-Euclidean kernels are employed to escape the exponential growth of the typical hypercube search space in the dimensionality of the problem. Cylindrical kernels [61, 43] transform the geometry of the search space, which consequently expands the center of the search space, shrinking the boundaries.

The three pieces of related work that are most similar to ours are Elastic GPs [47], SAASBO [15] and BOCK [43], which all perform optimization in the full-dimensional search space. Of these, [47] consider various lengthscales of the GP when optimizing the acquisition function, but uniquely does not impose simplifying assumptions on the model. [15] and BOCK [43] employ their aforementioned assumptions to facilitate effective optimization. Contrary to these works, we facilitate optimization in the ambient space without making any of the specific structural assumptions outlined in Sec. 3.

4 Pitfalls of High-Complexity Assumptions

We now discuss the issues related to highly complex models and connect it to the high-dimensional setting. Sec. 4.1 demonstrates the intuitive relation between complexity and dimensionality. Building upon the intuitive understanding of the MIG and the related model design choices gained in Sec. 2.3, we delve into the BO-specific pathology that arises from an overly complex model in Sec. 4.2, proving that it is distinct from the well-known *boundary issue* [60]. Thereafter, we demonstrate how various HDBO methods circumvent the high-complexity-issue by showing the how conventional structural assumptions reduce the model complexity. In subsequent sections, we will use the terms MIG and complexity interchangeably.

4.1 Complexity and Dimensionality

Increased model complexity most often becomes a critical issue for BO algorithms in high-dimensional problems — with increasing dimensions, the maximal space between points increases. Specifically, the expected distance between randomly sampled points in a unit cube increases proportional to the square root of the dimension [31]. For both the RBF and the Matern-$\frac{5}{2}$ kernels, this greatly impacts

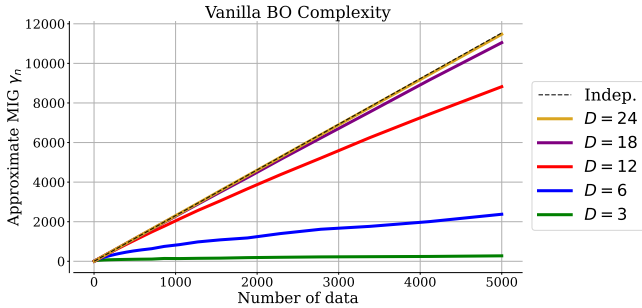

Fig. 6.2: Complexity scaling in the number of data points for varying dimensionalities of the problem for vanilla BO with a lengthscale of $\ell = 0.5$. For $D = 18$, the complexity visually differs from an independent kernel after approximately 3000 data points. For $D = 24$, 5000 data points are not sufficient to rid independence between observations. The MIG is approximated by sampling evenly distributed data using a SOBOL sequence.

the covariance, which decreases exponentially with the $\boldsymbol{\ell}$-normalized squared distance.

In Fig. 6.2, we display the scaling of the complexity in the number of data points, considering an RBF-kernel with fixed lengthscales. The curves represent increasing dimensionalities. As the dimensionality increases, the covariance matrix increasingly resembles that of an independent kernel (black dashed line). For $D = 18$ (purple), this manifests in a visible difference first after 3000 samples, whereas for $D = 24$ (yellow), 5000 samples is insufficient to distinguish an RBF kernel from an independent one, which implies that $k(\boldsymbol{X}_{5000}, \boldsymbol{X}_{5000}) \approx \boldsymbol{I}$. This strongly suggests that global modelling of the objective is uninformative, as the model quickly reverts back to its prior mean and variance even after collecting vast amounts of data, and meaningful inference between observed data points becomes very difficult.

4.2 The Boundary Issue Revisited

As covered in Sec. 2.3, high complexity implies that k produces relatively low correlation both between acquired data points and in prospective queries. Thus, the GP will only be informative close to existing observations, and will quickly revert back to the prior as we move away from this data, as demonstrated in the rightmost model in Fig. 6.1. Under this regime, the BO data acquisition will be highly dependent on the hyperparameters that dictate said GP prior, namely the signal variance σ_f^2 and the mean constant c. We note that, while these parameters are not always learned [11], the choice to fix them to 1.0 and 0.0 respectively influences on the behavior of the BO algorithm as well.

Introduced by [60], the boundary issue is the phenomenon that EI will, in high-dimensional settings, repeatedly query uninformed, high-variance points along the boundary of the search space to maximally explore in light of an uninformed model. We contrast this claim by the following proposition, which demonstrates that EI does *not* tend towards maximal variance when the model is uninformed, namely when $\mathbf{K} \approx \boldsymbol{I}$ (as in the $D = 24$ example in Fig. 6.2). We denote by $\boldsymbol{x}_{\text{inc}}$ the location of the incumbent, its value by $y_{\max}$, and the GP mean function by c.

Proposition 2 (**Lower Bound on EI Correlation**). *Assume that $y_{max} > c$, $\mathbf{K} = \sigma_f^2 \boldsymbol{I}$ and that the candidate query $\boldsymbol{x}_*$ correlates with at most one observation. Then, the correlation $\rho^* = \sigma_f^{-2} k(\boldsymbol{x}_*, \boldsymbol{x}_{inc})$ between the next query $\boldsymbol{x}_* = \arg\max_{\boldsymbol{x} \in \mathcal{X}} EI(\boldsymbol{x})$ and $\boldsymbol{x}_{inc}$ satisfies*

$$\rho^* \sqrt{\frac{1+\rho^*}{1-\rho^*}} \geqslant \frac{y_{max} - c}{\sigma_f}. \tag{6.3}$$

The proof parametrizes EI by the correlation ρ between a candidate $\boldsymbol{x}_*$ and the incumbent, and shows that $\frac{\partial \texttt{EI}}{\partial \rho}$ is positive for all values of ρ below the bound in Eq. 6.3. Thus, EI will prefer an observation that has substantial correlation with $\boldsymbol{x}_{inc}$ to one that does not.

Proposition 2 demonstrates that, when correlation in the model is low, EI does not seek out high-variance regions as described by [60]. This contrasts the common belief that HDBO intrinsically suffers from excessive exploration around the borders [53, 43, 16, 15, 38, 7, 14]. As we will soon cover, uninformative models frequently display the opposite behavior, where repeated queries are made exceedingly close to the incumbent.

In Fig. 6.3a, we observe this query behavior in action for the high-complexity model in Fig. 6.1 — despite many large-variance regions, the next query is very close to the current best, and within our correlation bound. In Fig. 6.3b, we display the numerical solution to $\frac{\partial \texttt{EI}}{\partial \rho} = 0$ together with the analytical bound in Proposition 2. We observe that for typical values of the GP mean and outputscale, the candidate query has substantial correlation with the incumbent under the aforementioned setting of an uninformed model.

With that said, the phenomenon of frequent querying of the boundary may still occur in practice . Specifically, querying of the boundary may occur when lengthscales are very long along one or more dimensions, which can occur when fitting the model using MLE [67, 48, 29]. Then, candidates will be low-variance and highly correlated with existing data, despite being located at the boundary of one or multiple dimensions that are all deemed irrelevant.

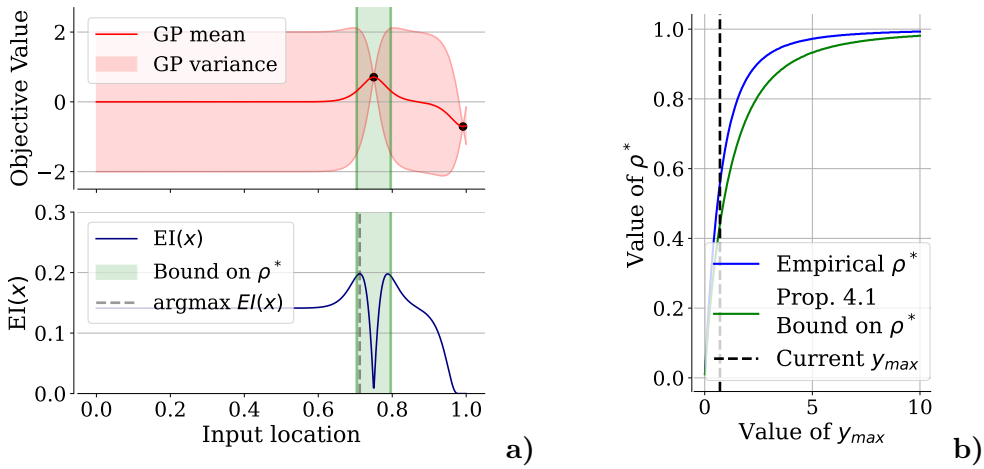

Fig. 6.3: Lower bound on the optimal correlation ρ^* between the incumbent and the upcoming query. **a)** The GP for two almost-independent observations with a large exploratory region. `EI` prefers to query close to the incumbent, well within the bound on ρ^* from Prop. 2. **b)** Tightness of the bound compared a numerical solve for optimal correlation for various values of y_{max}.

Proposition 2 establishes that `EI` does not intrinsically pursue high-variance, uninformed regions. Rather, queries preferentially have substantial correlation with the incumbent. In a high-complexity setting, substantial correlation only arises when a data point is close to existing data, which suggests that `EI` should make queries in close proximity to the incumbent. As a result, the algorithm tends to very seldom query far from the incumbent, resulting in an exploitative behavior with similar qualities to local search. This phenomenon arises when there is negligible correlation in the data, namely when model complexity is too high to effectively model the objective function with existing data.

4.3 Complexity of Existing HDBO

Having established that high complexity can yield uninformative models, as well as having discussed the link between complexity and dimensionality, it is evident that complexity assumptions must be sensible to facilitate a calibrated GP in HDBO. Notably, all the classes of HDBO algorithms outlined in Sec. 3 have such complexity-lowering assumptions. In Fig. 6.4, we display the modeled complexity of the most common classes of HDBO algorithms. Fig. 6.4 can be viewed as a cross section of Fig. 6.2, where we fix the number of data points to 1000 and instead vary the dimensionality of the problem to demonstrate how each HDBO class lower the growth of the complexity in the dimensionality, relative to a common

global GP model as well as an independent kernel. The methods presented are: REMBO [65] with $d_e = 4$, random AddGPs [70], BOCK [43], Local GPs [16] after one round of shrinkage, the global GP with fixed lengthscales from Fig. 6.2, and our proposed method — scaling the lengthscales in the dimensionality of the problem, which is introduced later in Sec. 5. While each algorithm has parameters that affect the MIG, we have set parameters to make the comparison as fair as possible.

As we have observed previously, employing a full-dimensional GP without restrictions is far too complex, as we have approximate independence after 1000 observations already for an $18D$-objective. As expected, random subspace methods have small complexity increase in the ambient dimensionality. The increase stems from the fact that random, non-axis-aligned embeddings may slice the ambient dimensions very narrowly, which results in shorter lengthscales on the embedded model. The complexity increase for both our method (blue) and cylindrical kernels (yellow) stagnates rapidly, effectively assuming only marginal complexity increases after $D = 100$. Local methods (red) scale at same rates as global GPs (blue), but work on a drastically simplified model due to the lengthscale-scaled trust regions.

We re-iterate that low complexity is not strictly a desirable property, but as per Occam's razor, the most desirable property is to have the lowest possible complexity for a model that sufficiently aligns with the objective. This is especially true in the context of small data optimization, where each new data point acquired and employed to train the model is costly. We note, however, that the almost-independence exhibited by the global GP in Fig. 6.4 (magenta) for even moderate dimensionalities inevitably leads to the degeneracy highlighted in Sec. 4.2. Moreover, the assumptions behind each HDBO method are all means to the same end - reducing model complexity to manageable levels.

5 Low-complexity High-dimensional Bayesian Optimization

Hypothesizing that the pitfalls of HDBO are strictly caused by assumptions of insurmountable complexity, we present our main methodological contribution. We design a simple, plug-and-play assumption that retains almost constant complexity as the dimensionality increases. Similar to Fig. 6.1, we achieve this by adjusting the prior on the lengthscales to the dimensionality of the problem to the task at hand. Moreover, we ensure a calibrated signal variance by drawing on previous findings on GPs in an over-parameterized regime.

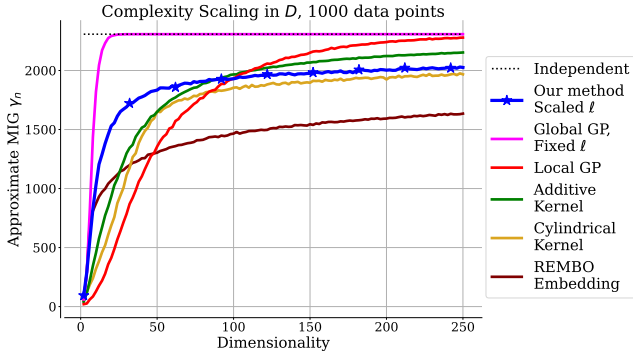

Fig. 6.4: We display the model complexity scaling in the dimensionality of the problem for 1000 data points for various HDBO algorithms. Vanilla BO with fixed lengthscales (magenta) approaches independent complexity at approximately 20 dimensions. As expected, REMBO random embeddings (brown) reduce complexity the most, followed by BOCK cylindrical kernels (yellow). The MIG growth of our proposed modification of the global GP (blue) flattens out at a rate similar to cylindrical kernels (yellow), despite modelling the original, full-dimensional space.

5.1 Ensuring Meaningful Correlation

Since stationary kernels compute covariances based on distances between data and both the diagonal and the distance between randomly sampled points in a D-dimensional hypercube scales as $\sqrt{D}$ [31], increasing the lengthscales at this rate, $\ell_i \propto \sqrt{D}$, counteracts the complexity increase that stems from the increased distances. This change may, for example, be achieved by scaling the μ term of a LogNormal ($\mathcal{LN}$) prior

$$\ell_i \sim \mathcal{LN}\left(\mu_0 + \frac{\log(D)}{2}, \sigma_0\right) \tag{6.4}$$

where (μ_0, σ_0) are suitable parameters of $p(\boldsymbol{\ell})$ for a one-dimensional objective. This shifts both the mode and mean of the distribution by a factor of $\sqrt{D}$. Notably, our method does not increase the number of hyperparameters in a MAP-based BO setup. Furthermore, the proposed change may similarly be applied the more commonly used Gamma prior, with a different parameterization [10]. Importantly, the change in complexity is not definitive, as we may still find some variables to be more important than others and adjust on-the-fly through MAP estimation of $\boldsymbol{\ell}$.

The proposed change suggests that the problem of large distances between points, and thereby the insurmountable complexity, is one that arises by assumption.

Specifically, the lengthscale priors $p(\boldsymbol{\ell})$ employed by conventional BO frameworks [3, 55, 24] place substantial density on low values of $\boldsymbol{\ell}$. Assuming that all dimensions are of major importance may appear like a conservative and sensible choice. For moderately high dimensions, however, it practically guarantees that the problem will be impossible to model globally, even with the largest of budgets. Our method takes the opposite approach, and simply assumes that a problem *is simple enough to be modeled globally, for any dimensionality.*

5.2 Calibrating Epistemic Uncertainty

Lastly, we consider the role of the signal variance parameter, whose impact on data acquisition is evidenced by Prop. 2. Motivated by findings on on the optimal value $\hat{\sigma}_f^2$ of σ_f^2 generally in [37] and in the over-parameterized regime by [42], we consider

$$\hat{\sigma}_f^2 = \frac{1}{n} \boldsymbol{y}^{\mathrm{T}} \mathbf{K}^{-1} \boldsymbol{y} \tag{6.5}$$

which, in a BO context, has a different impact than in GPs generally. As we are able to selectively acquire our data, a large number of parameters and substantially correlated data will simplify data fit, driving down the optimal value of σ_f^2. When data is repeatedly re-normalized, the issue will be reinforced, as the signal variance is further decreased, and another highly correlated query is selected. As such, we fix $\sigma_f^2 = 1$ to match the scale of the standardized observations, and to ensure that σ_f^2 does not diminish over time.

6 Results

We now compare our Vanilla Bayesian Optimization method with a dimensionality-scaled lengthscale prior (we will refer to our method as D-scaled $p(\boldsymbol{\ell})$ or DSP for clarity), against state-of-the-art HDBO methods. We will include various classes of HDBO methods, such as the subspace-methods Bounce and SAASBO [46, 15], the Local BO algorithms TuRBO [16] and Maximal Probability of Descent [41] (MPD), the AddGP method RD-UCB [70], the variable selection method MCTS-VS [57], and CMA-ES [22]. We use each method's official repository, with the exception of SAASBO which is run through Ax [2]. Our code is publicly available at `https://github.com/hvarfner/vanilla_bo_in_highdim`.

We instantiate the DSP with $\mu_0 = \sqrt{2}, \sigma_0 = \sqrt{3}$, which equates to $\boldsymbol{\ell} \approx 0.50$ for $D = 6$ under the mode of $p(\boldsymbol{\ell})$. We initialize all methods with 30 samples, marked

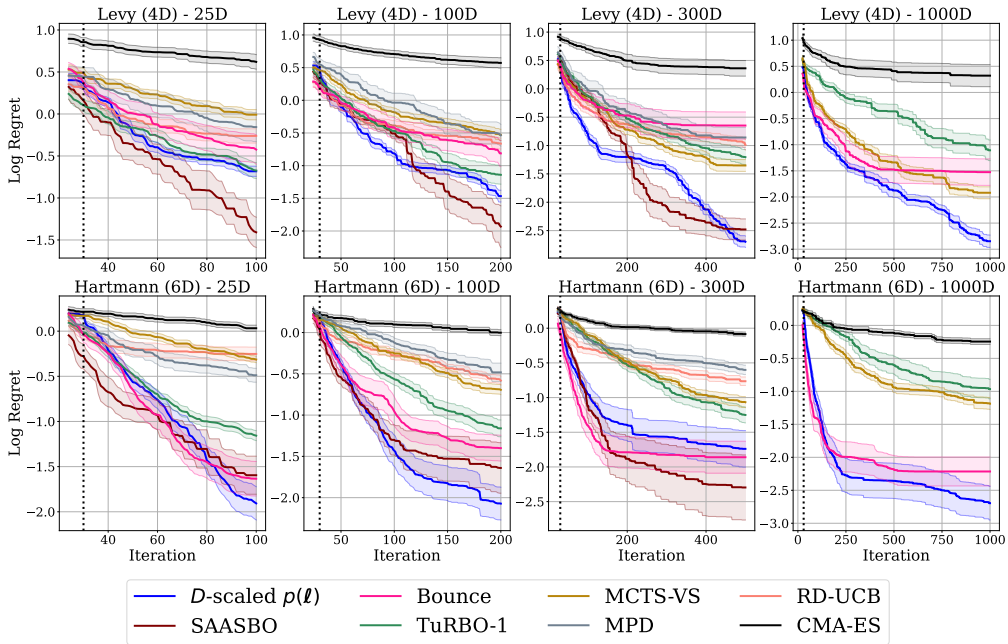

Fig. 6.5: Average log regret of all baselines on Levy (4D) and Hartmann (6D) synthetic test functions of varying ambient dimensionality across 20 repetitions (10 for SAASBO). Vanilla BO performs second best, beaten only by SAASBO on four tasks, whose axis-aligned subspace assumption (along with MCTS-VS' variable selection) aligns perfectly with the task at hand. We omit SAASBO from the 1000D benchmarks due to the prohibitive runtime, and RD-UCB and MPD due to a combination of runtime and numerical instability.

by a dashed vertical line. Bounce, CMA-ES and MPD deviate from conventional initialization. On all benchmarks, we use `LogEI` [1], using a low acquisition optimization budget of 512 initial (global) SOBOL samples and 512 Gaussian samples around the incumbent, followed by L-BFGS on the 4 best candidates, which is made possible by the low-complexity-high-smoothness model.

6.1 Sparse Synthetic Test Functions

We start by evaluating the DSP on a collection of commonly considered synthetic test functions with varying *total* and *effective dimensionality*. We note that the assumptions made in Sec. 5 diametrically oppose these test cases - each function has a low number of highly important dimensions with the large remainder being unimportant, whereas we assume that *each* dimension has relatively small impact.

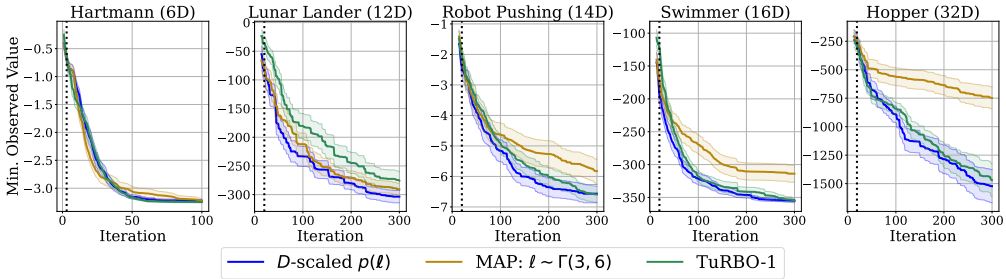

Fig. 6.6: Best observed value for the DSP, conventional MAP, and TuRBO on Hartmann (6D) and four mid-dimensional real-world optimization tasks. All methods perform comparably on Hartmann, while the DSP outperforms or is on part with TuRBO on the other tasks. Notably, the DSP performs at least equally well as the $\Gamma(3, 6)$ on all tasks, and substantially better on 4 out of 5 tasks, which suggests that it is well-suited as a drop-in replacement for conventional priors.

The DSP is highly performant, as it rapidly identifies the important dimensions and subsequently optimizes the task. This is similar in to [15], whose assumptions, represented through its sparse lengthscale prior, aligns perfectly with the task at hand. As such, SAASBO should, and does, perform best on average, with Vanilla BO being second. Notably, as Vanilla BO does not require the HMC [6] fully Bayesian model fitting that SAASBO uses, it runs in a small fraction of the time.

6.2 A Plug-in on Mid-Dimensional Tasks

Subsequently, we use the DSP as a plug-in for low- and mid-dimensional tasks, primarily those considered in [16], to evaluate its ability to serve as a substitute for conventional, non-adaptive hyperparameter priors. The Lunar Lander (12D) and Robot Pushing (14D) tasks from [66], as well as the Swimmer (16D) and Hopper (32D) reinforcement learning tasks from the MuJoCo suite [62], where we aim to learn a linear policy for two objects with varying degrees of freedom. We evaluate against a $\Gamma(3, 6)$ lengthscale prior with learnable σ_f^2, and against TuRBO, commonly considered the state-of-the-art mid-dimensional BO method. In Fig. 6.6, it is shown that the DSP is either competitive with, or outperforms, TuRBO on all tasks. The DSP maintains a moderate distance between queries, which indicates a calibrated trade-off throughout optimization, whereas the conventional $\Gamma(3, 6)$ does not.

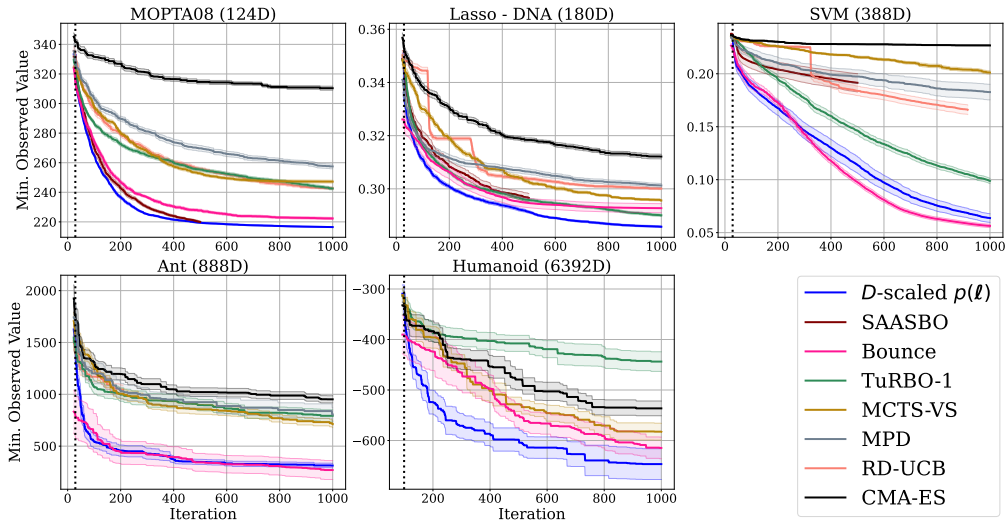

Fig. 6.7: Best observed value of all baselines on five real-world tasks from of various domains across 20 repetitions (10 for SAASBO). Vanilla BO performs best across all tasks except for MOPTA, where it is outperformed by SAASBO, and the MuJoCo Ant, where BAxUS gets an advantage from performing the initialization phase in a lower-dimensional subspace, which enables it to obtain initial samples close to the center of the search space. Vanilla BO substantially outperforms all baselines on Lasso-DNA, SVM and Humanoid. Notably, the extreme dimensionality of Humanoid does not have an apparent negative impact the performance of Vanilla BO.

6.3 High-dimensional Optimization Tasks

We now benchmark Vanilla BO with the DSP against a collection of frequently considered tasks in the high-dimensional literature [15, 16, 45, 46, 52]: Specifically, we consider MOPTA08 (124D), SVM (388D), Lasso-DNA (180D), and the MuJoCo [62] Ant (888D) and Humanoid (6392D) reinforcement learning tasks. We stress that, for all benchmarks where applicable, (BAxUS on SVM, Lasso-DNA and Humanoid, TuRBO on MOPTA and SVM, RD-UCB on Lasso-DNA), baselines perform within the error bars of the original implementation [45, 70] or in other papers by the same authors in the case of TuRBO [16, 15]. Fig. 6.7 shows that Vanilla BO with the DSP is highly competitive, performing the best by a substantial margin on Lasso-DNA and Humanoid, and produces top-two performance on the remaining tasks. On the MuJuCo Ant, Bounce's low-dimensional initialization allows it to obtain an average value of 800 after DoE due to consistently sampling data points close to the center of the search space. Notably, the DSP is very consistent between repetitions, as evident by the small error bars. This can be attributed to the consistent modelling, as the DSP

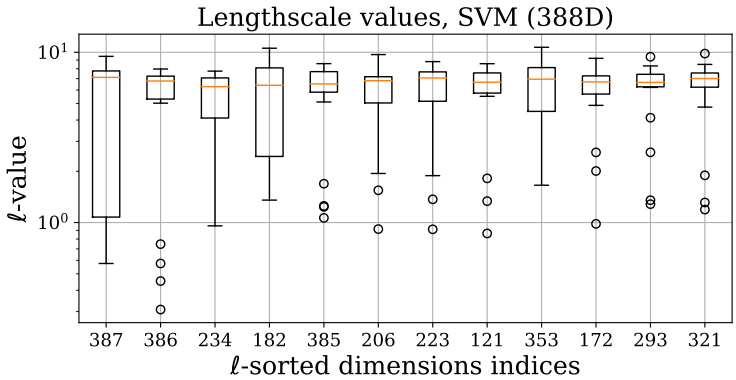

Fig. 6.8: Distribution of lengthscale values for the DSP on the 388D SVM task. Lengthscales are sorted according to their mean log value. The three last indexed dimensions (385, 386, 387) are considered particularly active [15], but are not identified as such consistently in our method. the black horizontal lines indicate upper and lower quartiles, and the orange horizontal lines indicate the median.

is not dependent on randomness in subspace design, trust region initialization or variable selection. Rather, it obtains a consistent, calibrated model through meaningful correlation in the data, from which it can effectively infer promising regions and improve upon the DoE.

Notably, the DSP does *not* heavily depend on identification of active variables. In Fig. 6.8, we demonstrate for the distribution of lengthscale values for the 388D SVM after 250 iterations. The DSP does not consistently identify active dimensions with large confidence. Instead, the calibrated complexity of the model allows for meaningful inference along all dimensions, until particularly active dimensions are potentially identified. As such, the DSP does not *require* the identification of active variables to achieve calibrated BO, but its identification helps optimization. As such, we attribute the superior performance of our method to the calibrated complexity, and the effective inference and exploration-exploitation trade-off that stems from it.

7 Conclusion and Future Work

The curse of dimensionality has long been assumed to hinder the application of conventional Bayesian optimization in high dimensions. We show that the hindrance is not driven by dimensionality, but by the assumed complexity of the objective. We make minor modifications to the assumptions of the vanilla BO

algorithm to make complexity scaling manageable with increasing dimensionality. As a result, we demonstrate that vanilla BO is extremely effective for problems of high dimensionality, outperforming the state-of-the-art for problems with dimensions into the thousands.

Nevertheless, we do not believe that tailored high-dimensional BO algorithms are unwarranted: if the problem at hand is known to adhere to the structural assumptions that are conventionally made (effective subspace, additivity) or where non-stationarity in the objective facilitates local modelling, we believe these approaches will be superior to the vanilla algorithm. However, these restrictive assumptions should not be made *out of necessity*, but when prior knowledge supports them. For future work, we plan to investigate the topic of complexity as it relates to modelling in Bayesian optimization more broadly, and in the context of latent space GP models [20, 36].

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

Carl Hvarfner, Erik Hellsten and Luigi Nardi were partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Luigi Nardi was partially supported by the Wallenberg Launch Pad (WALP) grant nbr. 2021.0348.

References

[1] Sebastian Ament, Sam Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for bayesian optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1vyAG6j9PE.

[2] Eytan Bakshy, Lili Dworkin, Brian Karrer, Konstantin Kashin, Benjamin Letham, Ashwin Murthy, and Shaun Singh. AE: A domain-agnostic platform for adaptive experimentation. In *Conference on Neural Information Processing Systems*, pages 1–8, 2018.

[3] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. In *Advances in Neural Information Processing Systems*, 2020. URL http://arxiv.org/abs/1910.06403.

[4] Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 462–471. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/baptista18a.html.

[5] Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. No-regret bayesian optimization with unknown hyperparameters. *Journal of Machine Learning Research*, 20(50):1–24, 2019. URL http://jmlr.org/papers/v20/18-213.html.

[6] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 2018.

[7] Mickaël Binois and Nathan Wycoff. A survey on high-dimensional gaussian process modeling with application to bayesian optimization. *ACM Trans. Evol. Learn. Optim.*, 2(2), aug 2022. doi: 10.1145/3545611. URL https://doi.org/10.1145/3545611.

[8] Mickaël Binois, David Ginsbourger, and Olivier Roustant. On the choice of the low-dimensional domain for global optimization via random embeddings. *J. of Global Optimization*, 76(1):69–90, jan 2020. ISSN 0925-

5001. doi: 10.1007/s10898-019-00839-1. URL https://doi.org/10.1007/s10898-019-00839-1.

[9] Adam D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12:2879–2904, 2011.

[10] Hye-Kyung Cho, Kenneth P. Bowman, and Gerald R. North. A Comparison of Gamma and Lognormal Distributions for Characterizing Satellite Rain Rates from the Tropical Rainfall Measuring Mission. *Journal of Applied Meteorology*, 43(11):1586–1597, November 2004. doi: 10.1175/JAM2165.1.

[11] George De Ath, Jonathan E. Fieldsend, and Richard M. Everson. What do you mean? the role of the mean function in bayesian optimisation. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, GECCO '20, page 1623–1631, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371278. doi: 10.1145/3377929.3398118. URL https://doi.org/10.1145/3377929.3398118.

[12] Josip Djolonga, Andreas Krause, and Volkan Cevher. High-dimensional gaussian process bandits. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/8d34201a5b85900908db6cae92723617-Paper.pdf.

[13] David K Duvenaud, Hannes Nickisch, and Carl Rasmussen. Additive gaussian processes. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/4c5bde74a8f110656874902f07378009-Paper.pdf.

[14] Afonso Eduardo and Michael U Gutmann. Bayesian optimization with informative covariance. *Trans. Mach. Learn. Res.*, 2023, 2022. URL https://api.semanticscholar.org/CorpusID:251320397.

[15] David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 493–503. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/eriksson21a.html.

[16] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/6c990b7aca7bc7058f5e98ea909e924b-Paper.pdf`.

[17] J. Gardner, C. Guo, K. Weinberger, R. Garnett, and R. Grosse. Discovering and Exploiting Additive Structure for Bayesian Optimization. In A. Singh and J. Zhu, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 1311–1319. Proceedings of Machine Learning Research, 2017.

[18] Roman Garnett, Michael A. Osborne, and Philipp Hennig. Active learning of linear embeddings for gaussian processes. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, page 230–239, Arlington, Virginia, USA, 2014. AUAI Press. ISBN 9780974903910.

[19] The GPyOpt-authors. GPyOpt: A bayesian optimization framework in python. `http://github.com/SheffieldML/GPyOpt`, 2016.

[20] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design. *arXiv: Machine Learning*, 2017.

[21] Eric Han, Ishank Arora, and Jonathan Scarlett. High-dimensional bayesian optimization via tree-structured additive models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):7630–7638, May 2021. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16933`.

[22] N. Hansen. The CMA evolution strategy: a comparing review. In J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer, 2006.

[23] Erik Orm Hellsten, Carl Hvarfner, Leonard Papenmeier, and Luigi Nardi. High-dimensional bayesian optimization with group testing, 2023.

[24] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*, 2011.

[25] Carl Hvarfner, Erik Hellsten, Frank Hutter, and Luigi Nardi. Self-correcting bayesian optimization through bayesian active learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=dX9MjUtP1A.

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

[27] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

[28] K. Kandasamy, J. Schneider, and B. Póczos. High Dimensional Bayesian Optimisation and Bandits via Additive Models. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, volume 37, pages 295–304. Omnipress, 2015.

[29] Toni Karvonen and Chris J. Oates. Maximum likelihood estimation in gaussian process regression is ill-posed. *Journal of Machine Learning Research*, 24(120):1–47, 2023. URL http://jmlr.org/papers/v24/22-1153.html.

[30] Johannes Kirschner, Mojmir Mutny, Nicole Hiller, Rasmus Ischebeck, and Andreas Krause. Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3429–3438. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kirschner19a.html.

[31] Mario Köppen. The curse of dimensionality. *5th online world conference on soft computing in industrial applications (WSC5)*, 2000.

[32] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(8):235–284, 2008. URL http://jmlr.org/papers/v9/krause08a.html.

[33] Ben Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional Bayesian optimization. *Advances in neural information processing systems*, 33:1546–1558, 2020.

[34] Cheng Li, Sunil Gupta, Santu Rana, Vu Nguyen, Svetha Venkatesh, and Alistair Shilton. High dimensional bayesian optimization using dropout. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 2096–2102. AAAI Press, 2017. ISBN 9780999241103.

[35] Mohit Malu, Gautam Dasarathy, and Andreas Spanias. Bayesian optimization in high-dimensional spaces: A brief survey. In *2021 12th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–8, 2021. doi: 10.1109/IISA52424.2021.9555522.

[36] Natalie Maus, Haydn Jones, Juston Moore, Matt J Kusner, John Bradshaw, and Jacob Gardner. Local latent space bayesian optimization over structured inputs. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 34505–34518. Curran Associates, Inc., 2022.

[37] C. J. Moore, A. J. K. Chua, C. P. L. Berry, and J. R. Gair. Fast methods for training gaussian processes on large datasets. *Royal Society Open Science*, 3(5):160125, May 2016. ISSN 2054-5703. doi: 10.1098/rsos.160125. URL `http://dx.doi.org/10.1098/rsos.160125`.

[38] Sarah Müller, Alexander von Rohr, and Sebastian Trimpe. Local policy search with bayesian optimization. In *Advances in Neural Information Processing Systems*, 2021.

[39] Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*, pages 4752–4761. PMLR, 2019.

[40] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions–i. *Math. Program.*, 14 (1):265–294, dec 1978. ISSN 0025-5610. doi: 10.1007/BF01588971. URL `https://doi.org/10.1007/BF01588971`.

[41] Quan Nguyen, Kaiwen Wu, Jacob Gardner, and Roman Garnett. Local bayesian optimization via maximizing probability of descent. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 13190–13202. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/555479a201da27c97aaeed842d16ca49-Paper-Conference.pdf`.

[42] Sebastian W. Ober, Carl E. Rasmussen, and Mark van der Wilk. The promises and pitfalls of deep kernel learning. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1206–1216. PMLR, 27–30 Jul 2021. URL `https://proceedings.mlr.press/v161/ober21a.html`.

[43] ChangYong Oh, Efstratios Gavves, and Max Welling. BOCK : Bayesian optimization with cylindrical kernels. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3868–3877. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/oh18a.html.

[44] Michael A Osborne. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, Oxford University, UK, 2010.

[45] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Increasing the scope as you learn: Adaptive bayesian optimization in nested subspaces. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=e4Wf6112DI.

[46] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Bounce: a Reliable Bayesian Optimization Algorithm for Combinatorial and Mixed Spaces. In *Advances in Neural Information Processing Systems*, 2023.

[47] Santu Rana, Cheng Li, Sunil Gupta, Vu Nguyen, and Svetha Venkatesh. High dimensional Bayesian optimization with elastic Gaussian process. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2883–2891. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/rana17a.html.

[48] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[49] Paul Rolland, Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. High-dimensional bayesian optimization via additive models with overlapping groups. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 298–307. PMLR, 09–11 Apr 2018. URL https://proceedings.mlr.press/v84/rolland18a.html.

[50] Jonas Rothfuss, Vincent Fortuin, Martin Josifoski, and Andreas Krause. Pacoh: Bayes-optimal meta-learning with pac-guarantees. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9116–9126. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/rothfuss21a.html.

[51] Jonas Rothfuss, Dominique Heyn, Jinfan Chen, and Andreas Krause. Meta-learning reliable priors in the function space. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[52] Kenan Šehić, Alexandre Gramfort, Joseph Salmon, and Luigi Nardi. LassoBench: A High-Dimensional Hyperparameter Optimization Benchmark Suite for Lasso. *arXiv preprint arXiv:2111.02790*, 2021.

[53] Eero Siivola, Aki Vehtari, Jarno P Vanhatalo, Javier I. González, and Michael Riis Andersen. Correcting boundary over-exploration deficiencies in bayesian optimization with virtual derivative sign observations. *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2017. URL `https://api.semanticscholar.org/CorpusID:53236252`.

[54] J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian optimization of machine learning algorithms. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems (NeurIPS'12)*, pages 2960–2968, 2012.

[55] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, page 2951–2959, Red Hook, NY, USA, 2012. Curran Associates Inc.

[56] J. Snoek, K. Swersky, R. Zemel, and R. Adams. Input warping for Bayesian optimization of non-stationary functions. In E. Xing and T. Jebara, editors, *Proceedings of the 31th International Conference on Machine Learning, (ICML'14)*, pages 1674–1682. Omnipress, 2014.

[57] Lei Song, Ke Xue, Xiaobin Huang, and Chao Qian. Monte carlo tree search based variable selection for high dimensional bayesian optimization. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=SUzPos_pUC`.

[58] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*, pages 1015–1022. Omnipress, 2010.

[59] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, May 2012. ISSN 1557-9654. doi: 10.1109/tit.2011.2182033. URL http://dx.doi.org/10.1109/TIT.2011.2182033.

[60] K. J. Swersky. *Improving Bayesian Optimization for Machine Learning using Expert Priors*. PhD thesis, University of Toronto, 2017.

[61] Kevin Swersky, David Duvenaud, Jasper Snoek, Frank Hutter, and Michael A. Osborne. Raiders of the lost architecture: Kernels for bayesian optimization in conditional parameter spaces, 2014.

[62] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

[63] Xingchen Wan, Vu Nguyen, Huong Ha, Binxin Ru, Cong Lu, and Michael A. Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10663–10674. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/wan21b.html.

[64] Linnan Wang, Rodrigo Fonseca, and Yuandong Tian. Learning search space partition for black-box optimization using monte carlo tree search. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19511–19522. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e2ce14e81dba66dbff9cbc35ecfdb704-Paper.pdf.

[65] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

[66] Zi Wang, Chengtao Li, Stefanie Jegelka, and Pushmeet Kohli. Batched high-dimensional Bayesian optimization via structural kernel learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3656–3664. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/wang17h.html.

[67] Christopher Williams and Carl Rasmussen. Gaussian processes for regression. In D. Touretzky, M.C. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995. URL https://proceedings.neurips.cc/paper_files/paper/1995/file/7cce53cf90577442771720a370c3c723-Paper.pdf.

[68] Kaiwen Wu, Kyurae Kim, Roman Garnett, and Jacob R. Gardner. The behavior and convergence of local bayesian optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=9KtX12YmA7.

[69] Yuling Yao, Aki Vehtari, and Andrew Gelman. Stacking for non-mixing bayesian computations: The curse and blessing of multimodal posteriors, 2020.

[70] Juliusz Krzysztof Ziomek and Haitham Bou Ammar. Are random decompositions all we need in high dimensional Bayesian optimisation? In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 43347–43368. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ziomek23a.html.